

VISOKA TEHNIČKA ŠKOLA U BJELOVARU

ZAVRŠNI RAD br: 24/MEH/2015

**Upravljanje pametnom kućom pomoću
mbed mikroračunala**

Matija Galjer

Bjelovar, travanj 2016.

VISOKA TEHNIČKA ŠKOLA U BJELOVARU

ZAVRŠNI RAD br: 24/MEH/2015

**Upravljanje pametnom kućom pomoću
mbed mikroračunala**

Matija Galjer

Bjelovar, travanj 2016.



Visoka tehnička škola u Bjelovaru

Trg E. Kvaternika 4, Bjelovar

1. DEFINIRANJE TEME ZAVRŠNOG RADA I POVJERENSTVA

Kandidat: **Galjer Matija**

Datum: 07.09.2015.

Matični broj: 000488

JMBAG: 0314004816

Kolegij: **MIKRORAČUNALA**

Naslov rada (tema): **Upravljanje pametnom kućom pomoću mbed mikroračunala**

Mentor: **Zoran Vrhovski, mag.ing.el.techn.inf.**

zvanje: **predavač**

Članovi Povjerenstva za završni rad:

1. dr.sc. Igor Petrović, , predsjednik
2. Zoran Vrhovski, mag.ing.el.techn.inf., mentor
3. Božidar Hršak, mag.ing.mech, član

2. ZADATAK ZAVRŠNOG RADA BROJ: 24/MEH/2015

U radu je potrebno opisati projektiranje i upravljanje pametne kuće. U uvodnom djelu završnog rada opisati koncept pametne kuće. Opisati i izraditi maketu pametne kuće s rasvjetom, grijačim tijelima, pomičnim roletama i zaključavanjem vrata. Opisati upravljanje pametne kuće pomoću mbed mikroračunala i izraditi softver u programskom okruženju mbed. Napraviti i opisati aplikaciju u programskom alatu APP Inventor kojom će se na daljinu upravljati pametnom kućom pomoću mbed mikroračunala.

Zadatak uručen: 07.09.2015.

Mentor: **Zoran Vrhovski, mag.ing.el.techn.inf.**



Sadržaj

Uvod.....	1
1. Općenito o pametnim kućama	2
1.1 X10 protokol.....	2
1.2 CEBus protokol.....	3
1.3 Konnex protokol.....	4
1.4 ZigBee protkol.....	4
2. Izrada makete pametne kuće	6
3. Upravljačka ploča pametne kuće.....	8
3.1 Mbed mikroračunalo.....	8
3.2 Koračni motor.....	11
3.3 Grijač i ventilator.....	12
3.4 Elektromagnetska brava.....	12
3.5 Relejni modul	13
3.6 PIR senzor.....	13
3.7 Bluetooth.....	14
3.8 Ostale elektroničke komponente.....	16
4. Izrada upravljačke pločice pametne kuće.....	17
4.1 Shema spajanja drivera ULN2003.....	18
4.2 Shema spajanja relejnog modula	18
4.3 Shema spajanja PIR senzora.....	19
4.4 Shema spajanja Bluetooth modula.....	20
4.5 Ostale elektroničke komponente.....	21
5. Programski kod mbed mikroračunala.....	22
5.1 Program upravljanja pametnom kućom.....	24
5.1.1 Upravljanje osvjetljenjem.....	25
5.1.2 Upravljanje i regulacija temperature.....	25
5.1.3 Upravljanje vratima	25
5.1.4 Režimi rada.....	25
6. Izrada aplikacije za upravljanje pametnom kućom	27
6.1 Dizajner komponenata	27
6.2 Uređivač blokova	28
6.3 Testiranje aplikacije.....	29
6.4 Aplikacija za upravljanje pametnom kućom	31
7. Prikaz rada pametne kuće.....	35
7.1 Upravljanje osvjetljenjem modela pametne kuće	36
7.2 Prikaz regulacije temperature unutar modela pametne kuće.....	36

7.3	<i>Prikaz rada protuprovalne zaštite modela pametne kuće i upravljanje vratima.....</i>	<i>37</i>
8.	Zaključak.....	38
9.	Literatura	39
	Sažetak.....	42
	Abstract	43
	Privitak	46

Uvod

Razvojem tehnologije ljudima se pokušava stvoriti što jednostavniji i kvalitetniji život, kako van doma tako i unutar njega. Jedan od načina poboljšanja kvalitete življenja unutar doma ostvaruje se njegovom automatizacijom. Moderan princip takve automatizacije naziva se pametna kuća. Pomoću senzora, aktuatora, glavne upravljačke jedinice (mikroračunala) i android mobilnog uređaja zaduženog za interakciju između korisnika i kuće moguće je modernizirati vlastiti stil privatnog života.

Upravo je navedeni primjer glavna tema završnoga rada. Senzori koji se koriste u završnome radu su temperaturni senzor i senzor pokreta. Interakcija je ostvarena pomoću aplikacije dizajnirane u programskome alatu *App Inventor* tako da korisnik na što jednostavniji način zadaje naredbe *mbed* LPC1768 mikroračunalu koji upravlja aktuatorima kao što su LED diode, koračni motori, ventilator, grijač i elektromagnetska brava. Završni rad podijeljen je na sedam poglavlja u kojima se detaljno opisuje izrada modela pametne kuće, tiskane pločice, izvedba programskog koda i dizajniranje aplikacije. Također su opisani korišteni programski alati koji su olakšali konačnu izvedbu praktičnog dijela završnoga rada.

1. Općenito o pametnim kućama

Pojam pametne kuće poznat je desetljećima, a ideja je nastala zbog premalo odvojenog privatnog vremena, što uzrokuje neuredan i neorganizirani način življenja. Osnovni zadatak pametne kuće je osigurati udoban i siguran privatni život. Uključuje kvalitetno iskorištavanje slobodnog vremena i najbitnije uštedu energije. Napretkom na tržištu, proizvodnja hardverskih i softverskih tehnologija postaje sve jeftinija i pristupačnija. Također, automatizacija kućanstava se nadograđuje i širi svoje mogućnosti do te mjere da čovjek može upravljati kućom iz svoga naslonjača.

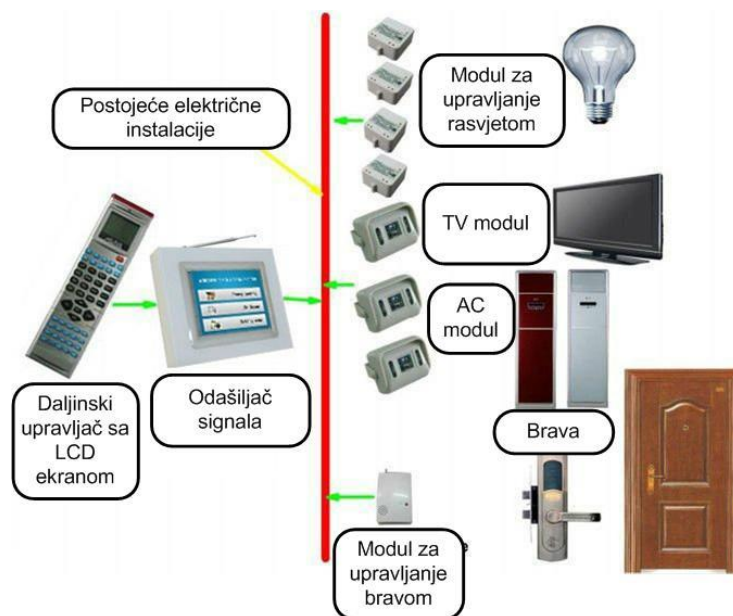
Sustav pametne kuće čini skup kućne automatike i sigurnosnog sustava. Takav sustav moguće je podijeliti na više podsustava zaduženih za obavljanje raznih funkcija kao što su upravljanje rasvjetom, klimatizacijom, grijanjem, vratima, roletama, funkcije protuprovalne zaštite te zaštite od poplave i požara. Osnovnu opremu sustava pametne kuće čine centralna jedinica (najčešće LCD ekran na dodir), mikromoduli, senzori, tipkala i daljinski upravljači. Centralna jedinica povezuje cjelokupnu kućnu automatiku. Omogućuje korisniku da nadgleda kuću pomoću kamera, ali i uz pomoć ugrađenog modema daje mogućnost upravljanja preko mobilnog telefona. Kod ugradnje ovakvih sustava postoje protokoli koji određuju pravila oko povezivanja uređaja unutar pametne kuće. Neki od poznatih protokola su:

1. *X10*,
2. *CEBus*,
3. *Konnex* i
4. *Zigbee*.

1.1 X10 protokol

X10 najstariji je protokol razvijen sedamdesetih godina prošlog stoljeća. Koristi postojeću infrastrukturu električne mreže za komunikaciju između električnih uređaja. Ožičenje koje služi za napajanje uređaja istovremeno se koristi za slanje digitalnog podatka ukomponiranog u val nosioc frekvencije 120 kHz. Podatak se sastoji od adrese i naredbe koja je poslana s upravljačke jedinice. Adresa je veličine osam bitova što znači da je moguće imati maksimalno 256 spojenih različitih uređaja ili više ako pojedini uređaji koriste istu adresu. Naredba je veličine četiri bita.

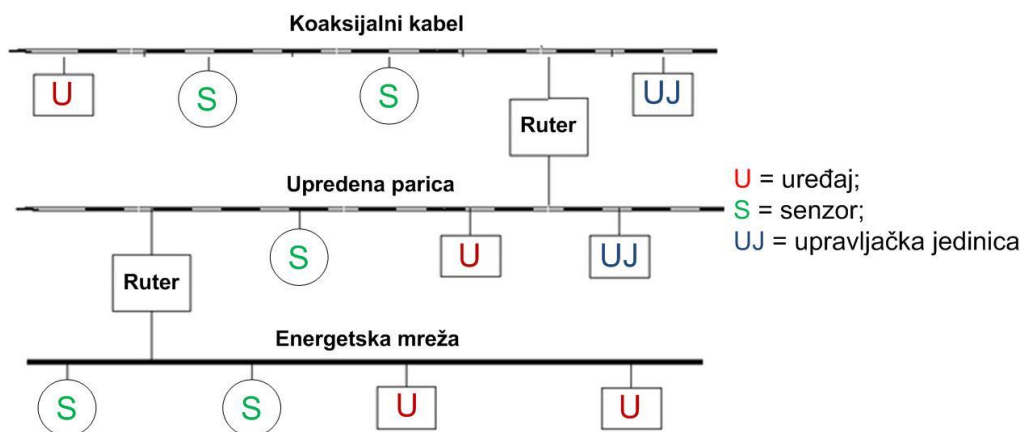
Zbog korištenja postojećih instalacija X10 protokol ima određenu prednost, ali i nedostatak u obliku smetnji koje mogu prouzrokovati kućanski aparati s elektromotorom. Smetnje se mogu eliminirati pomoću raznih filtera. Slika 1.1. prikazuje princip X10 protokola. Povezuju se uređaji preko modula koji se spajaju na postojeće električne instalacije [1].



Slika 1.1. X10 protokol [2]

1.2 CEBus protokol

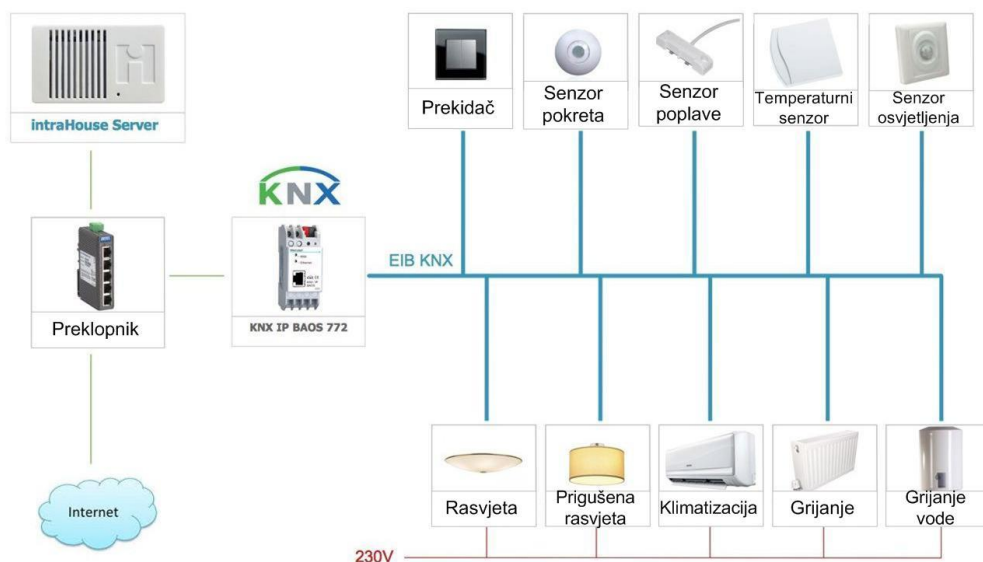
CEBus (eng. *Consumer Electronic Bus*) protokol nastao je kao poboljšanje tada jedinog postojećeg X10 protokola. Tražila su se rješenja oko drugih medija za prijenos informacija kao što su: koaksijalni kabel, gradska mreža, upletena parica, optička vlakna, infracrvena i radio frekventna komunikacija. Protokol koristi *peer-to-peer* topologiju mreže ,gdje svi uređaji osluškuju poruke istovremeno. Brzina prijenosa je oko 8 kbps [3]. Slika 1.2. prikazuje CEBus protokol.



Slika 1.2. CEBus protkol [4]

1.3 Konnex protokol

Godine 1997. nastaje najpoznatiji europski protokol *Konnex*. Ujedinjuje tri prijašnja najraširenija europska protokola BatiBUS, EIB (engl. *European Installation Bus*) i EHS (engl. *European Home Systems*) i postaje prvi međunarodno priznati standard u automatizaciji pametnih kuća. Za prijenos informacija *Konnex* protokol koristi iste medije kao i CEBus te dodatno još i *Ethernet*. Najčešće je izveden pomoću upletene parice preko koje se razmjenjuju informacije između centralne jedinice i senzora ili aktuatora. Upravljanje može biti izvedeno preko računala ili mikroračunala. Fleksibilnost je jedna od prednosti ovog protokola. Instalacije se lako prilagođavaju uređajima koji su naknadno dodani u sustav, odnosno spojeni na postojeću instalaciju [5]. Slika 1.3. prikazuje *Konnex* protokol.



Slika 1.3. *Konnex* protokol [6]

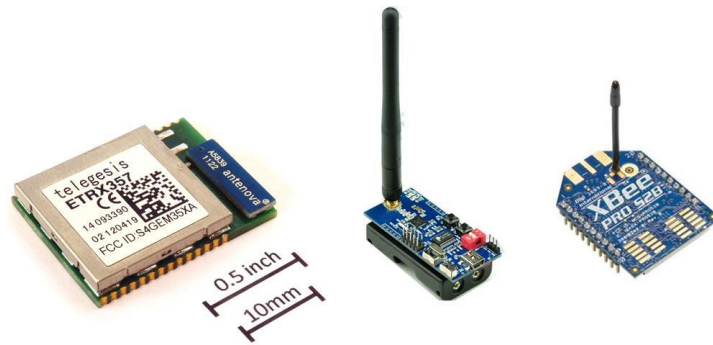
1.4 ZigBee protkol

ZigBee komunikacijski protokol baziran je na stvaranju osobne mreže s ciljem da se ostvari:

1. mala propusnost,
2. visoka sigurnost prijenosa informacije,
3. umrežavanje velikog broja uređaja i
4. mala potrošnja energije.

Bežične mreže poput *Bluetooth*-a ili *Wi-Fi*-a idealne su za ovu vrstu protokola zbog svoje male potrošnje čija je vrijednost proporcionalna udaljenosti (radijusu) upravljanja.

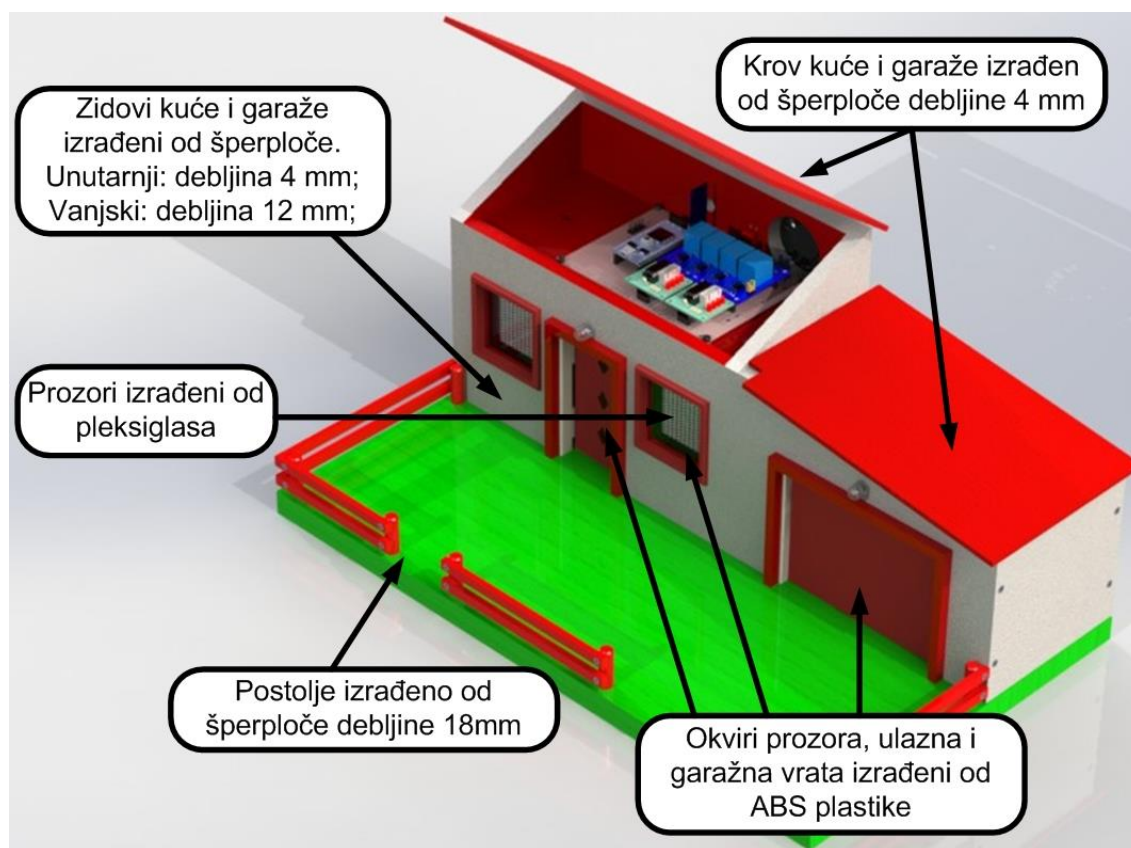
Protokol koristi razgranatu i zvjezdastu topologiju mreže. Slika 1.4. prikazuje *ZigBee* uređaj. Ovi su uređaji malih dimenzija, jeftini i pouzdani [7].



Slika 1.4. *ZigBee* uređaji [7], [8], [9]

2. Izrada makete pametne kuće

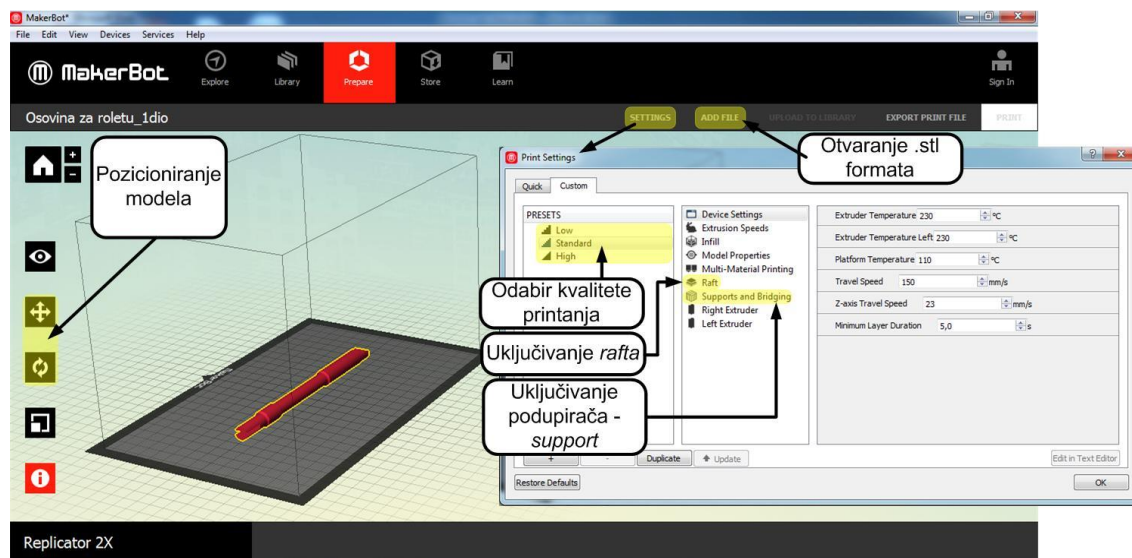
Prije izrade makete pametne kuće, potrebno je izvesti njen dizajn u nekome od CAD¹ alata kao što je *SolidWorks*. Programski alat *SolidWorks* povećava kvalitetu dizajna, precizniju izradu, a najbitnije smanjuje potrebno vrijeme od same ideje pa do konačne izrade gotovog proizvoda. Naziv pametna kuća se ne mora odnositi samo na stambene objekte, već i na spremišta, prostore gdje ljudi rijetko obitavaju kao što su garaže ili dvorišta. Maketa osim kuće, kao stambenog dijela sadrži i garažu kojom je moguće upravljati. Kuća je podijeljena na četiri prostorije (boravak, hodnik, spavaća soba i tavan) s vlastitom LED rasvjetom. Pretpostavka je da će se najviše obitavati u boravku, stoga su upravo tamo smješteni grijač i temperaturni senzor. Sa stražnje strane makete postavljen je otvor za ventilator. Prednja strana makete sadrži dva otvora za prozore boravka i spavaće sobe, dok je središnji otvor za ulazna vrata u hodnik. Opis iskorištenog osnovnog materijala za izradu makete prikazan je na slici 2.1.



Slika 2.1. Maketa pametne kuće [10],[11],[12]

¹ eng. *Computer-Aided Design*

Za izradu vrata kuće i garaže, kućišta ventilatora te osovine za podizanje i spuštanje roleta korišten je 3D printer. Postupak izrade 3D printanjem vrlo je jednostavan pošto se modeliranje izvršava također u *SolidWorksu*. Izrađeni model potrebno je spremiti u STL format prikladan za programski alat *MakerWare* (slika 2.2.).



Slika 2.2. Programski alat *MakerWare*

3. Upravljačka ploča pametne kuće

Nakon izrade makete, slijedi izrada upravljačkog dijela pametne kuće. Kada se spomene pametna kuća, prve poveznice na taj pojam su:

- regulacija temperature,
- upravljanje rasvjetom i osvjetljenjem unutar i izvan kuće,
- protuprovalna zaštita i
- ušteda energije.

Gore navedeni pojmovi predstavljaju podsustave pametne kuće završnog rada. Regulacija temperature ostvarena je grijačem, ventilatorom i temperaturnim senzorom. Rasvjetu i osvjetljenje čine LED diode i rolete koje pokreću koračni motori, dok je protuprovalna zaštita izvedena preko elektromagnetske brave i PIR senzora. Sve navedene aktuatora potrebno je povezati preko centralne upravljačke jedinice koju čini *mbed* mikroračunalo. *Mbed* mikroračunalo prikuplja podatke s ulaznih pinova te preko izlaznih pinova direktno ili indirektno upravlja aktuatorima.

3.1 Mbed mikroračunalo

Mikroračunala su sve zastupljenija u svijetu. Postaju sve moćnija, jeftinija i interaktivnija te se zbog toga počinju upotrebljavati u svim granama industrije, ali i u domaćinstvu. Pristupačna su za širi krug korisnika koji traže brži razvoj prototipova. Upravo to nude *mbed* mikroračunala vodećeg svjetskog poluvodičkog dobavljača ARM-a. ARM u suradnji s više od 200 kompanija (AMD, Intel, Apple, Samsung, Lenovo, ...) predstavlja „srce“ razvoja digitalnih poluvodičkih proizvoda. Kao upravljačka jedinica *mbed* mikroračunala koristi se mikrokontroler ARM CORTEX M3 LPC1768. *Mbed* mikroračunalo usmjereno je na brzu izradu prototipa (eng. *Rapid Prototyping*) što je pogodovalo kod izvedbe pametne kuće. Brza izrada prototipa uključuje DIL² oblik pakiranja *mbed* mikroračunala, razvijene API³ komponente te WEB stranice za međusobnu komunikaciju između razvojnih inženjera. Karakteristike LPC1768 mikrokontrolera su:

- ARM 32-bitni procesor od 100MHz,
- DIL podnožje s 40 pinova,

² eng. *Dual In Line* – integrirani krug kojeg čini pravokutno kućište sa dva reda nožica.

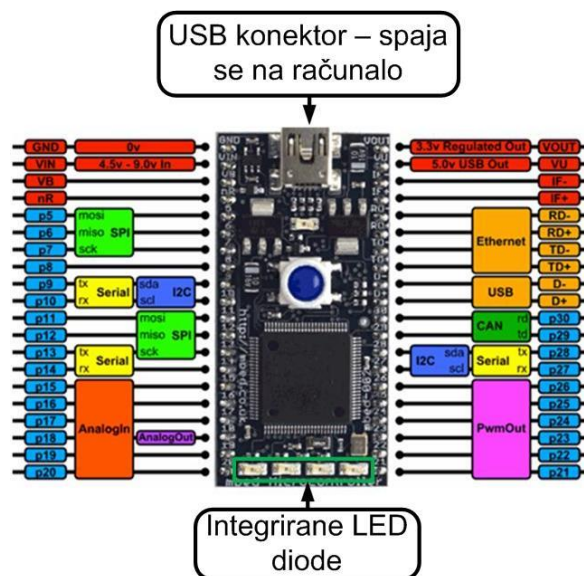
³ eng. *Application Programming Interface* – Aplikacijsko Programsko Sučelje. Standard kojeg svi programeri slijede pri programiranju.

- 512 kB programska *Flash* memorija,
- 2x32 kB SRAM memorija,
- sučelja (prikazana na slici 3.1.):
 - USB,
 - *Master/Slave SPI* (eng. *Serial Pheripheral Interface*),
 - *I²C* (eng. *Inter Integrated Circuit*),
 - CAN (eng. *Controller Area Network*),
 - serijsko sučelje i
 - Ethernet (10/100 MBit)
- osam kanala analogno-digitalnih pretvornika (rezolucija od 12 bitova),
- digitalno-analogni pretvornik (rezolucije 10 bitova),
- pulsno širinska modulacija (PWM) (šest kanala),
- osam DMA (eng. *Direct Memory Access*) kanala,
- UART (eng. *Universal Asynchronous Receiver and Transmitter*) četiri kanala,
- SSP⁴ dva kanala,
- jedan I²S⁵,
- četiri 32-bitna tajmera/brojača,
- RC oscilator od 4 MHz,
- radni napon: 4.5-9 V.

Na razvojnom okruženju *mbed* mikroračunala nalaze se četiri LED diode koje predstavljaju dodatna četiri digitalna izlaza. Napajanje *mbed* mikroračunala može biti izvedeno preko USB konektora ili može biti vanjsko iznosa 4.5 V do 9 V [13]. Sva navedena sučelja, raspored pinova i integriranih LED dioda prikazana su na slici 3.1. Za izradu završnoga rada korišteni su pinovi prikazani u tablici 3.1.

⁴ eng. *Service Switching Point* – poboljšana SPI komunikacija.

⁵ eng. *Integrated Interchip Sound* – serijsko sučelje namjenjeno za povezivanje audio uređaja zajedno.



Slika 3.1. *Mbed* LPC1768 [14]

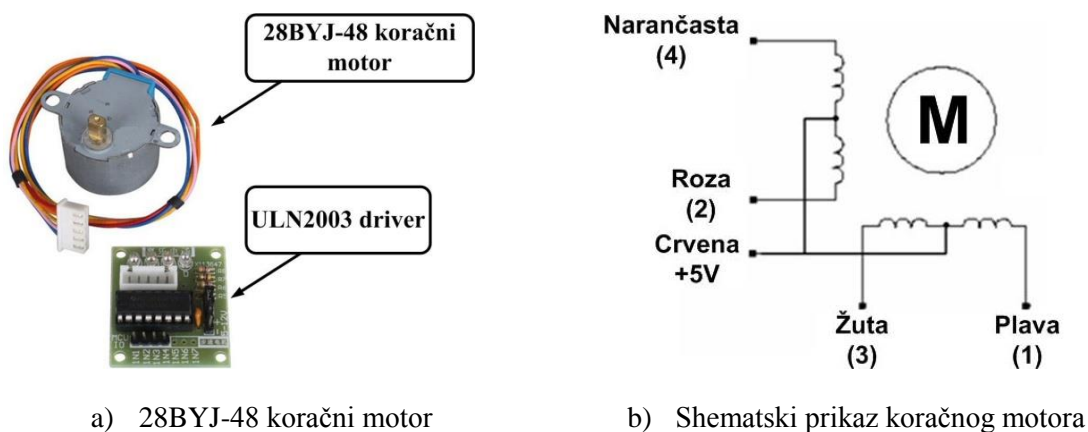
Tablica 3.1. Korišteni pinovi *mbed* mikroračunala

Pin/pinovi	Opis pinova
P5, P6, P8, P7	Upravljački pinovi IN4, IN3, IN1, IN2 koračnog motora za upravljanje roletama
P9, P10, P11, P12	Upravljački pinovi IN4, IN3, IN2, IN1 koračnog motora za upravljanje garažnim vratima
P13	RXD pin <i>Bluetooth</i> modula
P14	TXD pin <i>Bluetooth</i> modula
P15	Upravljački pin IN1 četverokanalnog relejnog modula za upravljanje ventilatorom
P16	Upravljački pin IN2 četverokanalnog relejnog modula za upravljanje grijačem
P17	Upravljački pin IN3 četverokanalnog relejnog modula za upravljanje elektromagnetskom bravom
P18	Izlazni pin LM35 temperaturnog senzora
P27	Izlazni pin PIR senzora
P26, P25, P24, P23, P22	Anode LED dioda
GND	GND, katode LED dioda
VOUT = 3.3 V	Napajanje VCC <i>Bluetootha</i>
VU = 5 V	Napajanje VCC PIR senzora, LM35 senzora

3.2 Koračni motor

Koračni motori posebna su vrsta istosmjernih motora. Upravljanjem zavojnicama motora zakreće se njegova osovina u koracima. U završnome radu funkciju podizanja i spuštanja roleta na kući, odnosno garažnih vrata obavlja unipolarni koračni motor 28BYJ-48 prikazan na slici 3.2.a).

Unipolarni koračni motori najčešće imaju dvije zavojnice podijeljene sa srednjim izvodom. Zbog podjele može se reći da zapravo ima četiri zavojnice. Pomicanje rotora koračnog motora u traženome smjeru vrši se propuštanjem struje određenim redoslijedom kroz zavojnice statora. Raspored zavojnica i shematski dio koračnog motora prikazani su na slici 3.2.b).



Slika 3.2. Koračni motor [15]

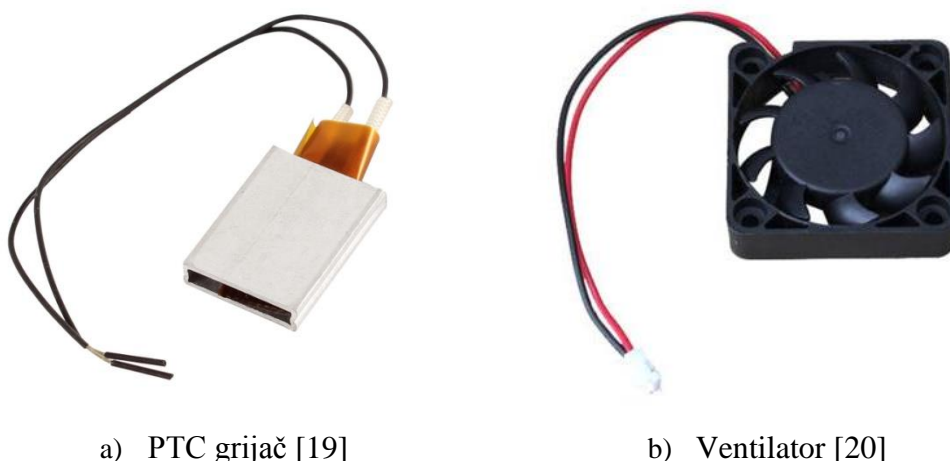
Sa slike 3.2 može se zaključiti da motor ima pet žica za spajanje; po dvije za svaku zavojnicu i jednu zajedničku za napajanje [16]. Izlaz mikroračunala ne može dati potrebnu struju za pogon koračnog motora, stoga se motor povezuje preko ULN2003 drivera. Specifikacije koračnog motora prikazane su u tablici 3.2.

Tablica 3.2. Specifikacije koračnog motora [17]

Model	28BYJ-48
Radni napon	5 V
Broj faza	4
Frekvencija	100 Hz
Broj koraka po jednome okretaju	64
Kut koraka	5,625°
Buka	<35 dB

3.3 Grijač i ventilator

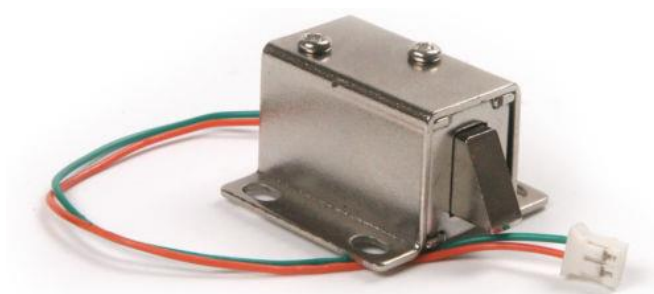
PTC⁶ grijač i ventilator dio su sustava regulacije temperature. Prednost PTC grijača je što se ponaša kao termostat, odnosno porastom temperature se povećava njegov otpor. Keramičkog je materijala smještenog u aluminijsko kućište [18]. Za hlađenje kuće korišten je računalni ventilator nazivnog napona 12 V. Slika 3.3. prikazuje grijač i ventilator korišteni za upravljanje temperaturom pametne kuće.



Slika 3.3. Grijah i ventilator

3.4 Elektromagnetska brava

Funkciju zaključavanja i otključavanja ulaznih vrata obavlja elektromagnetska brava upravljana preko releja. Vrata će biti otključana dovodenjem napona na ulazne stezaljke brave. Slika 3.4. prikazuje elektromagnetsku bravu korištenu za protuprovalnu zaštitu makete pametne kuća.

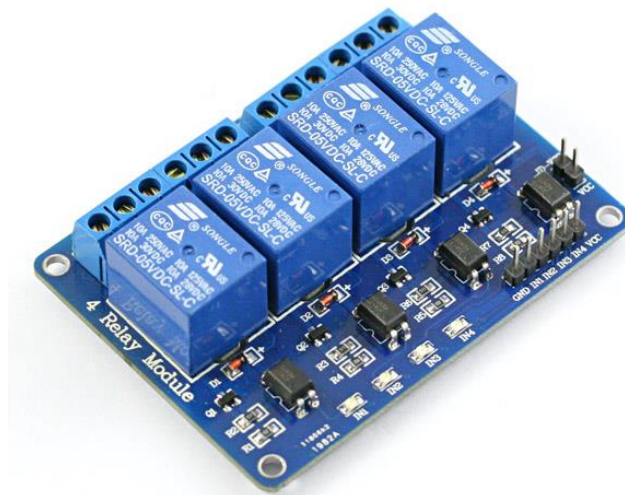


Slika 3.4. Elektromagnetska brava [21]

⁶ eng. *Positive Temperature Coefficient*

3.5 Relejni modul

Četverokanalni relejni modul s optokaplerima koristi se za uključivanje i isključivanje uređaja čija je nazivna struja veća od one koju može dati mikroračunalo. Dovođenjem električnog signala na pinove IN1 do IN4 aktivira se određeni relej. Optokapleri imaju ulogu električnog odvajanja strujnog kruga mikroupravljača od strujnog kruga elektromagneta releja. Pojedini relej ima tri izlazne stezaljke: NC (normalno zatvorenu), NO (normalno otvorenu) i zajedničku COM stezaljku. U slučaju normalno otvorenog kontakta (NO), kada se napon dovede na elektromagnet, on će privući kotvu i zatvoriti strujni krug. Kod normalno zatvorenog kontakta (NC) slučaj je obrnut. Slika 3.5. prikazuje četverokanalni relejni modul.



Slika 3.5. Četverokanalni relejni modul [22]

3.6 PIR senzor

Uloga senzora pokreta je detekcija uljeza unutar kuće. Sastavljen je od dva dijela osjetljiva na infracrveno zračenje. U stanju mirovanja oba dijela će detektirati istu količinu zračenja iz okruženja. Prolaskom toplog objekta ispred senzora najprije će reagirati jedna polovica te izazvati promjenu napona. Kada isti objekt napusti tzv. osjetilnu zonu senzora, doći će do negativne promjene napona. Slika 3.6. prikazuje PIR senzor.



Slika 3.6. Pir senzor [23],[24]

Specifikacije HC-SR501 senzora korištenog u završnom radu vidljive su u tablici 3.3. PIR senzor sadrži kratkospojnik kojim određujemo način okidanja nakon detekcije pokreta koje može biti neprekidno (označeno slovom H) ili samo jednom (označeno slovom L) [25]. Kako bi se poboljšala funkcionalnost, ali i imunost na vanjske smetnje poput vlage, senzor je hermetički zatvoren u metalno kućište.

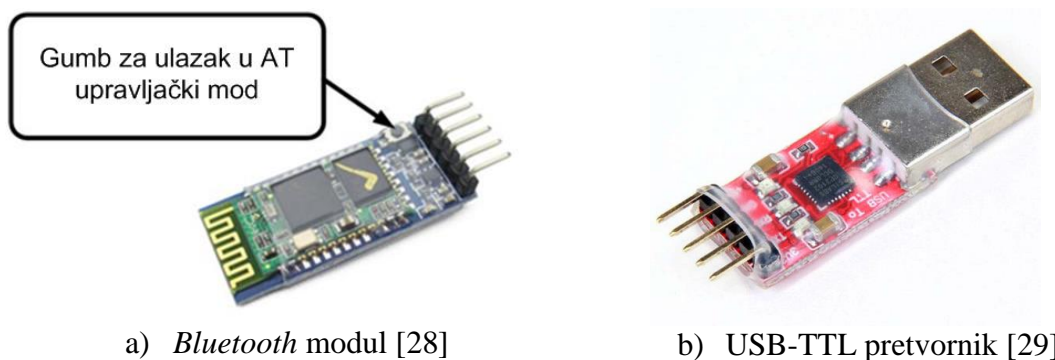
Tablica 3.3. Specifikacije pir senzora [26]

Model	HC-SR501
Radni napon	4.5 V ÷ 20 V
Razina napona na izlazu	Visoka razina = 3.3 V; niska razina = 0 V
Kut detekcije	<100°

3.7 Bluetooth

Bluetooth tehnologija vrsta je jeftine bežične komunikacije na kratkim udaljenostima s minimalnom potrošnjom energije. Domet *Bluetooth* veze kreće se do deset metara u krugu oko uređaja, ali s povećanjem snage odašiljača može dosegnuti i do stotinu metara. Veza koristi ISM⁷ nelicencirani frekvencijski pojas (od 2,4 GHz do 2,4835 GHz) [27]. Nelicencirano znači da nije potrebna dozvola za njeno korištenje te da korisnik ne mora plaćati naknadu za korištenje. Zbog cijene i prisutnosti u svakom mobilnom uređaju, *Bluetooth* se koristi kao dio komunikacije između korisnika (android mobilnog uređaja) i *mbed* mikroračunala kao upravljačkog dijela pametne kuće. Slika 3.7.a) prikazuje korišteni *Bluetooth* modul u završnome radu.

⁷ eng. *Industrial, Scientific and Medical*



Slika 3.7. *Bluetooth* modul i USB-TTL pretvornik

Mbed mikroračunalo ne može direktno komunicirati s androidom, već preko HC05 *Bluetooth* modula. Prije samog povezivanja *Bluetooth* modula s mikroračunalom, potrebno ga je parametrirati [30], odnosno:

1. odrediti brzinu prijenosa podataka (eng. *Baud rate*),
2. odrediti prepoznatljivo ime modula,
3. odrediti lozinku,
4. odrediti stop i paritetni bit za detekciju greške kod prijenosa i
5. odrediti ulogu modula (master ili slave).

Parametriranje *Bluetooth* modula izvršava se ulaskom u upravljački mod. Parametri se unose AT naredbama, a najčešće korištene naredbe prikazane su u tablici 3.4.

Tablica 3.4. Najčešće korištene AT naredbe [31],[32]

Naredba	Odgovor	Parametar	Opis naredbe
AT	OK	Nema	Provjera da li je uspostavljena komunikacija
AT+ORGL	OK	Nema	Vraćanje na tvorničke postavke (default)
AT+ADDR?	+ADDR: <Param> OK	Adresa bluetooth modula	Dohvat <i>Bluetooth</i> adrese
AT+NAME?	+NAME <Param> OK	Ime <i>Bluetooth</i> modula	Dohvat imena <i>Bluetooth</i> modula

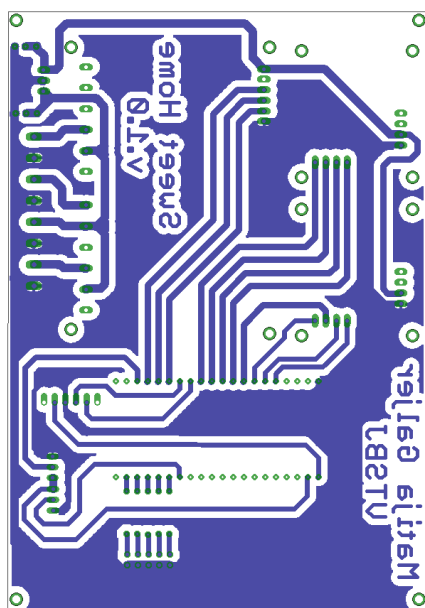
AT+NAME = <Param>	OK	Ime <i>Bluetooth</i> modula	Dodjeljivanje novog imena
AT+ROLE?	+ROLE:<Param>	0=slave; 1=master; 2=slave-loop	Upit koju ulogu ima modul
AT+UART?	+UART: <Param1>, <Param2>, <Param3>	Param1 = baud rate; Param2 = stop bit; Param3 = paritetni bit.	Dohvat parametara serijske komunikacije
AT+UART= <Param1>, <Param2>, <Param3>	OK	Param1 = baud rate; Param2 = stop bit; Param3 = paritetni bit.	Postavljanje parametara serijske komunikacije
AT+PSWD?	+PSWD: <Param>	Lozinka	Dohvat lozinke za pristup modulu
AT+PSWD= <Param>	OK	Lozinka	Upis lozinke

3.8 Ostale elektroničke komponente

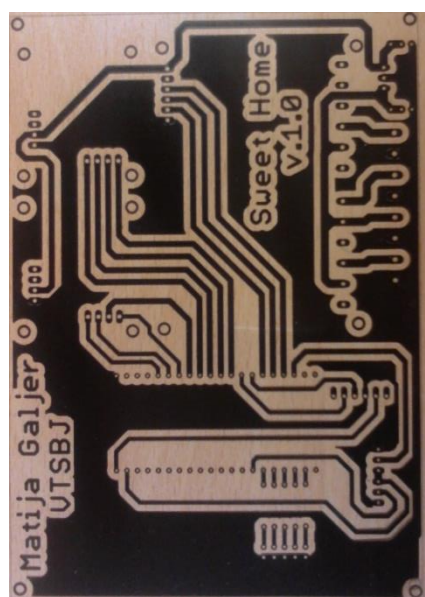
Pod ostale elektroničke komponente spadaju otpornici za ograničavanje struje na LED diodama i kondenzatori za filtriranje napona. Pretvorbu napona s 12 V na 5 V izvršava se pomoću stabilizatora 7805 pomoću kojeg se napaja relej i ULN2003 driveri koračnih motora. Kod sustava regulacije temperature potreban je temperaturni senzor LM35, a za jednostavnije spajanje komponenti na tiskanu pločicu korišteni su konektori i redne stezaljke.

4. Izrada upravljačke pločice pametne kuće

Radi profesionalnosti, preglednosti i rasporeda komponenti potrebno je bilo izraditi tiskanu pločicu. Komponente koje se nalaze na tiskanoj pločici su driveri (ULN2003) koračnih motora, relejni i *Bluetooth* modul (HC05), *mbed* mikroračunalo (LPC1768), stabilizator napona (7805), otpornici, kondenzatori, konektori i redne stezaljke. Dizajniranje pločice izvedeno je u programskome alatu *Eagle*. Za završni rad bilo je potrebno samostalno dizajnirati pojedine komponente (driver za koračni motor, relejni i *Bluetooth* modul) pošto nisu bile sadržane unutar biblioteke programskog alata *Eagle*.



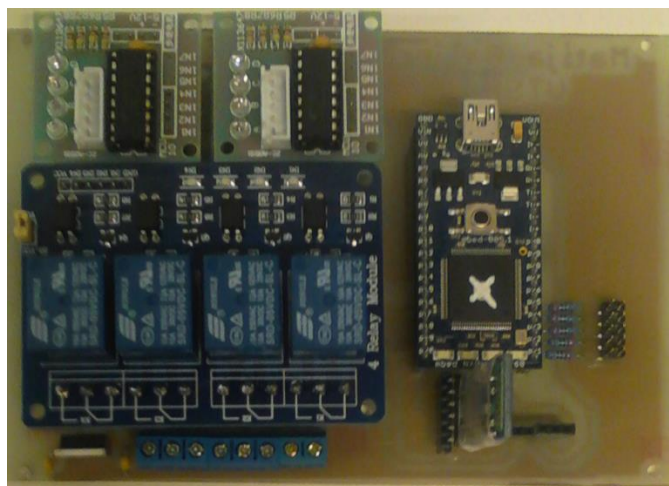
a) *Eagle* – raspored vodova i otočića



b) Ispisani film na foliji

Slika 4.1. Izrada tiskane pločice

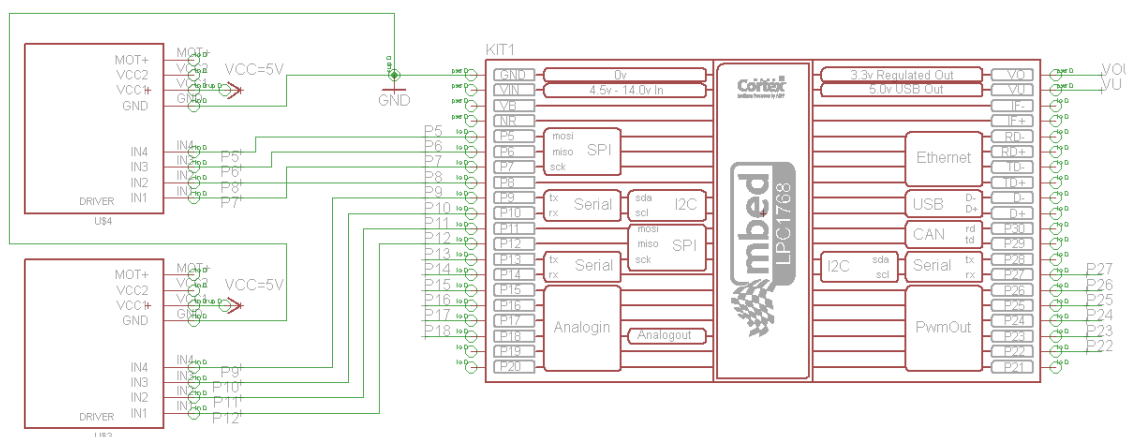
Projektirana pločica u programskome alatu *Eagle* (slika 4.1.a) korištena je za izradu filma koji je ispisan na prozirnu foliju (slika 4.1.b). Osvjetljivanjem pločice UV svjetlom preko folije uklanja se fotolak na mjestima gdje folija nije zatamnjena. Ostatak fotolaka se uklanja uranjanjem pločice u razvijatelj (natrijev hidroksid). Lak će ostati samo na vodovima. Mješavinom solne kiseline i hidrogena (jetkanjem) odstranjuje se sav bakar osim na vodovima na kojima je ostao fotolak. Nakon jetkanja tiskane pločice, uslijedila je zaštita vodova kositrenjem i bušenje rupa za montiranje elektroničkih komponenata na tiskanu pločicu [33]. Slika 4.2. prikazuje tiskanu pločicu sa spojenim komponentama.



Slika 4.2. Tiskana pločica sa spojenim komponentama

4.1 Shema spajanja drivera ULN2003

Upravljačka ploča na sebi ima dva drivera. Prvi driver koristi se za upravljanje koračnog motora za rolete, a drugi za upravljanje garažnim vratima. Pinovi IN1, IN2, IN3 i IN4 pojedinog koračnog motora spojeni su redom na digitalne ulaze P7, P8, P6 i P5, odnosno na P12, P11, P10 i P9 *mbed* mikroračunala. Napajanje drivera može se izvesti pomoću *mbed* mikroračunala ili pomoću 7805 stabilizatora napona. Slika 4.1. prikazuje shemu spajanja ULN2003 drivera na *mbed* mikroračunalo.

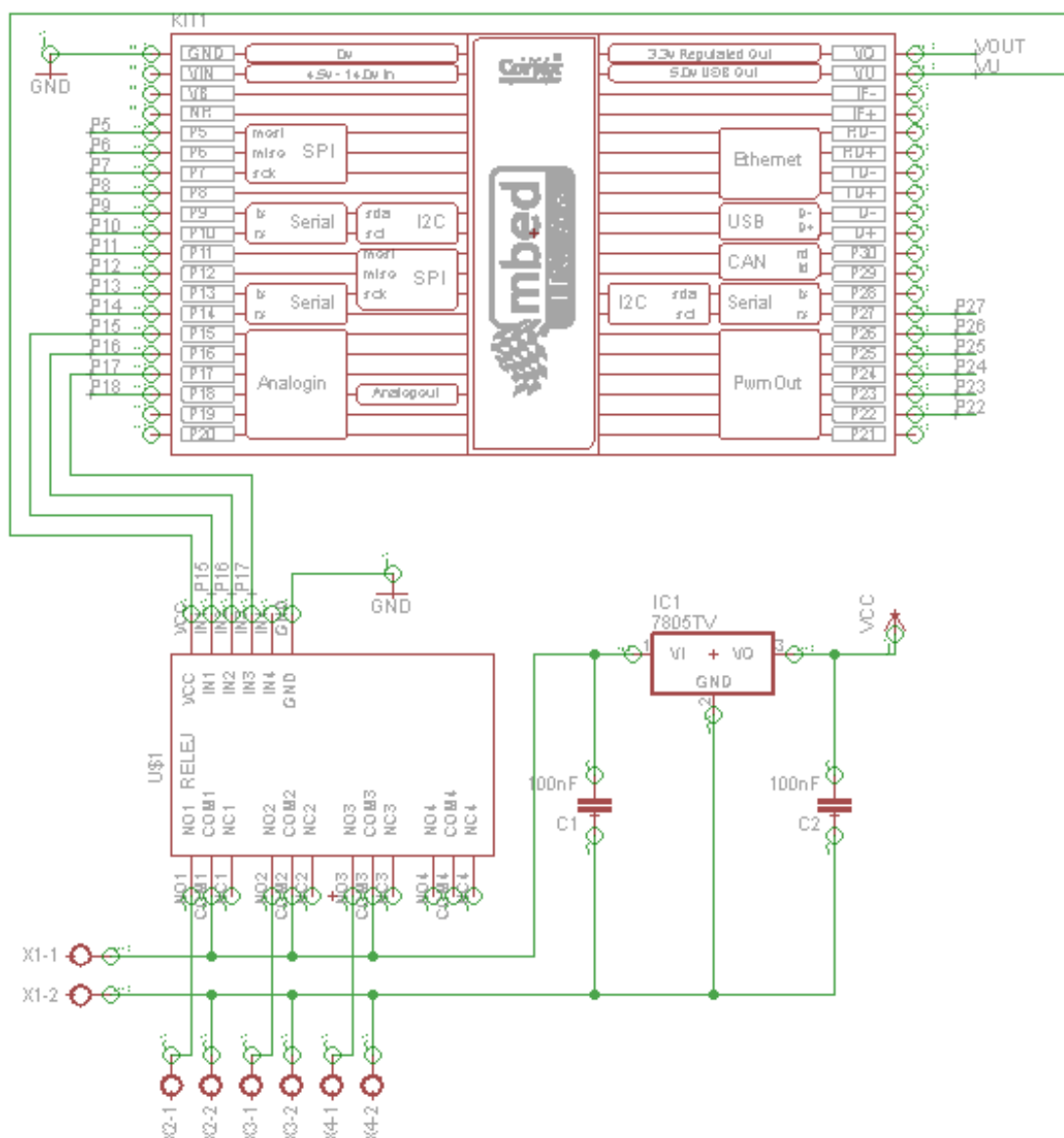


Slika 4.1. Shema spajanja ULN2003 drivera na *mbed* mikroračunalo

4.2 Shema spajanja relejnog modula

Relejni modul sadrži četiri releja pomoću kojih se upravlja ventilatorom, grijačem i bravom. Stabilizator napajanja 7805 korišten je za pretvorbu napona vanjskog izvora napajanja 12 V na 5 V potrebnih za napajanje relejnog modula i ULN2003 drivera.

Kondenzatori C1 i C2 (100 nF) uklanjaju naponske smetnje. Zajedničke COM stezaljke relejnog modula spojene su na pozitivnu nožicu napajanja X1-1. Negativna nožica napajanja (X1-2) redne stezaljke X1 spojena je s negativnim nožicama rednih stezaljki: X2, X3 i X4 preko kojih su spojeni grijač (X2-1), ventilator (X3-1) i brava (X4-1). Slika 4.2. prikazuje shemu spajanja relejnog modula na *mbed* mikroračunalo i stabilizator napona.

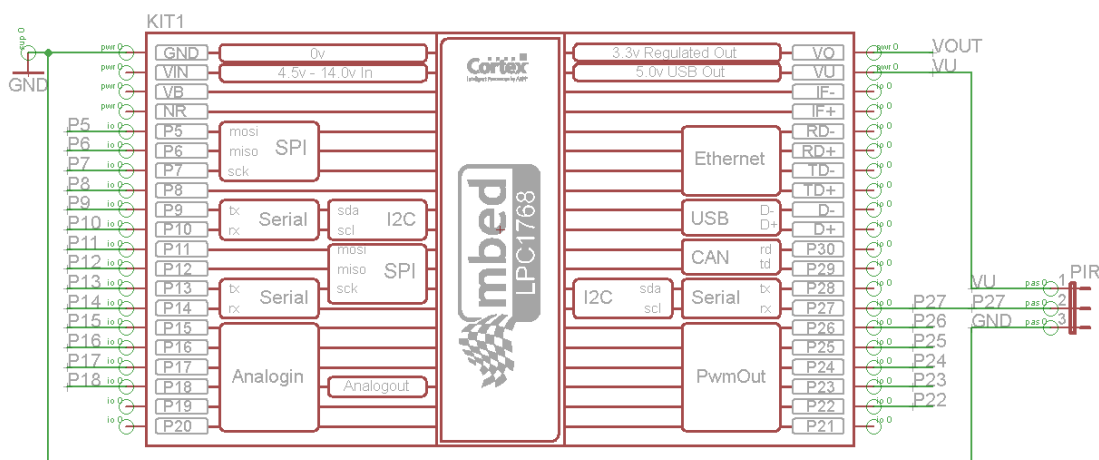


Slika 4.2. Shema spajanja relejnog modula na *mbed* mikroračunalo

4.3 Shema spajanja PIR senzora

PIR senzor, smješten unutar modela pametne kuće povezan je sa *mbed* mikroračunalom preko konektora smještenog na upravljačkoj pločici. Izlazni pin PIR senzora spojen je na analogni pin (P27) *mbed* mikroračunala. PIR senzor radi na naponu

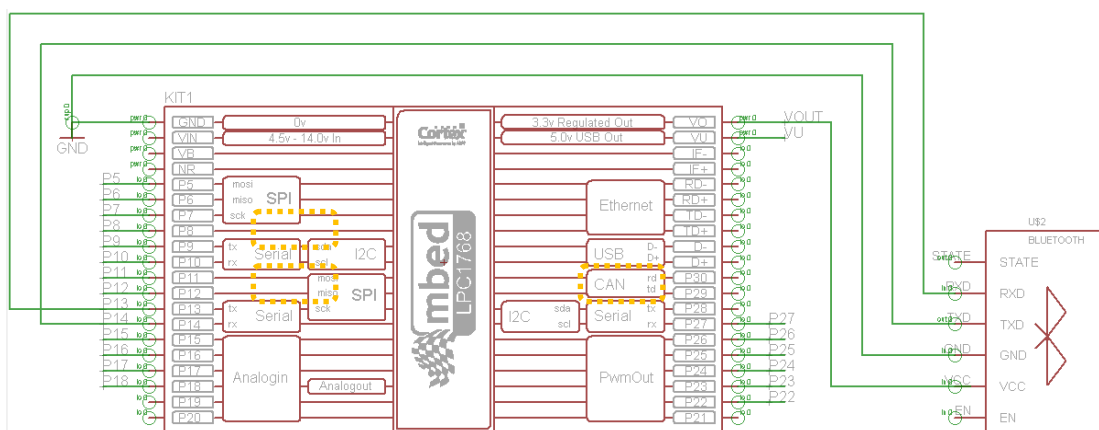
od 5 V dobivenog preko VU pina *mbed* mikroračunala. Slika 4.3. prikazuje shemu spajanja PIR senzora na *mbed* mikroračunalo.



Slika 4.3. Shema spajanja pir senzora na *mbed* mikroračunalo

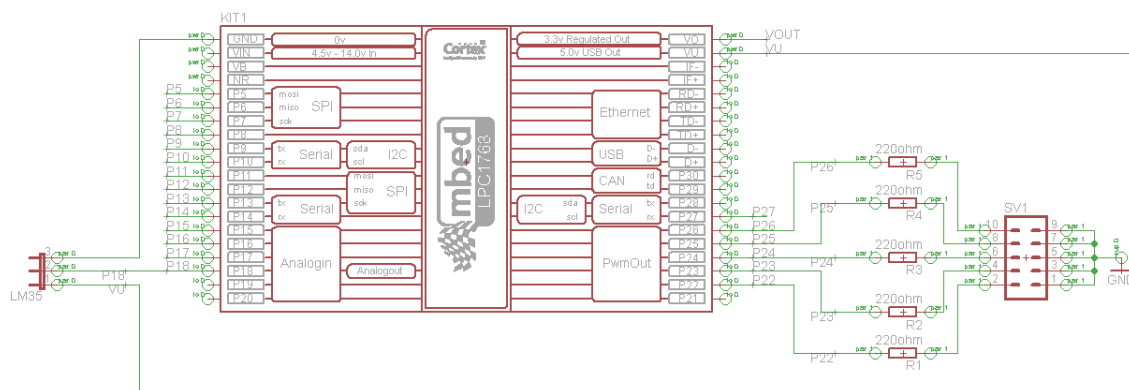
4.4 Shema spajanja *Bluetooth* modula

Bluetooth modul ima napajanje iznosa 3.3 V što je također izvedivo preko *mbed* mikroračunala (VOUT pin). *Mbed* mikroračunalo ima točno definirane pinove za takvu vrstu veze. Za komunikaciju *Bluetooth* modul ima dva pina. RXD za primanje podataka i TXD za slanje podataka, koji su unakrsno spojeni na RXD i TXD *mbed* mikroračunala. Slika 4.4. prikazuje shemu spajanja *Bluetooth* modula na serijsko sučelje *mbed* mikroračunala.



Slika 4.4. Shema spajanja bluetooth modula na *mbed* mikroračunalo

4.5 Ostale elektroničke komponente



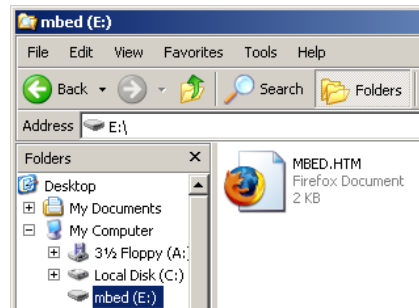
Slika 4.5. Shema spajanja LED dioda i LM35 senzora na *mbed* mikroračunalo

Od senzora još se koristi temperaturni senzor LM35 za potrebe regulacije temperature. Napajanje senzora kreće se između 4 V i 20 V stoga mu je najpajanje iznosa 5 V dovoljno za ispravan rad. LM35 ima tri pina. Izlazni pin daje analognu vrijednost napona proporcionalnog temperaturi okruženja. Stoga je izlaz temperaturnog senzora moguće spojiti samo na jedan od šest pinova (P15÷P20) *mbed* mikroračunala.

LED diode se ne nalaze na upravljačkoj ploči, ali ih je potrebno povezati sa *mbed* mikroračunalom preko konektora. Anode LED dioda spajaju se preko jednog reda nožica konektora na PWM izlaze (P22÷P26) *mbed* mikroračunala. PWM izlazi se koriste radi promjene intenziteta svjetlosti. Drugi red nožica povezuje katode svih LED dioda na zajednički potencijal GND. Slika 4.5. prikazuje shemu spajanja LED dioda i LM35 temperaturnog senzora.

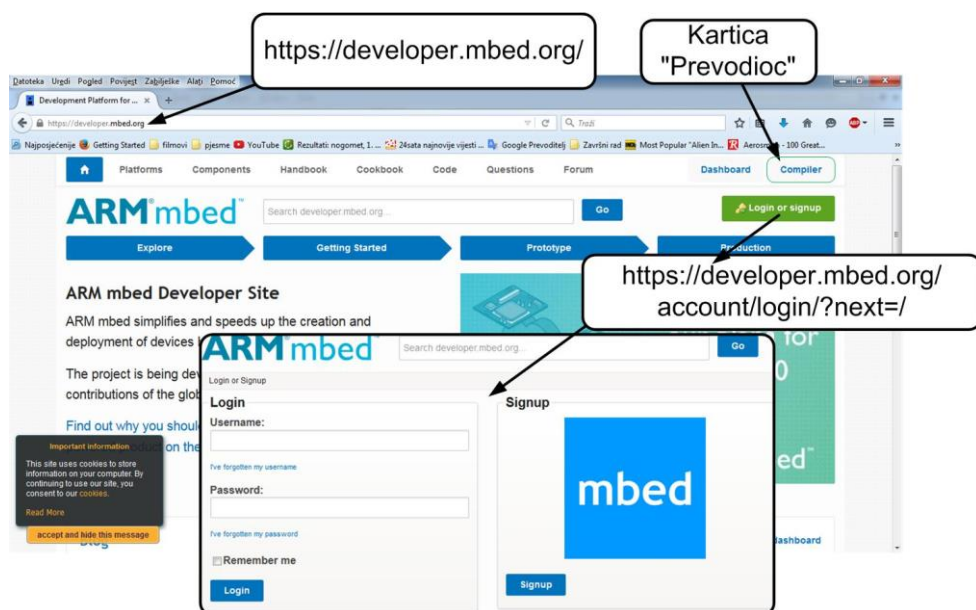
5. Programski kod mbed mikroračunala

Programiranje *mbed* mikroračunala započinje njegovim spajanjem na računalo, koje ga prepoznaje kao vanjsku *flash* memoriju tj. kao USB disk. Slika 5.1. prikazuje *mbed* kao USB *flash* memoriju.



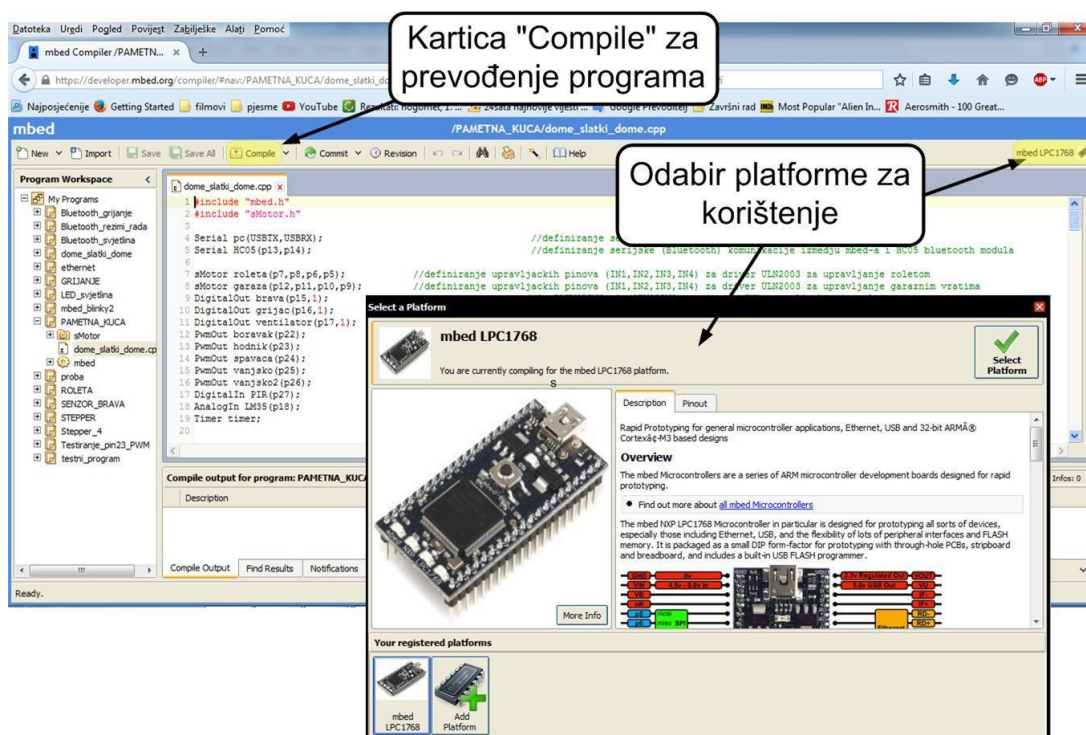
Slika 5.1. Mbed kao USB flash disk

Nakon što računalo prepozna *mbed* mikroračunalo, potrebno je otvoriti MBED.HTM datoteku ili pokrenuti *mbed* korisničku podršku na web stranici <https://developer.mbed.org>. Slijedi registracija vlastitog ili prijava postojećeg korisničkog računa. Prijavom, korisnik ostvaruje pristup svim postojećim sadržajima poput biblioteka ili programa koje su kreirali drugi korisnici. Također, web stranica sadrži forum gdje je moguće diskutirati oko problema vezanih za programiranje ili povezivanja *mbed* mikroračunala s vanjskim komponentama. Početna stranica *mbed* razvojnog okruženja i registracija prikazani su na slici 5.2.



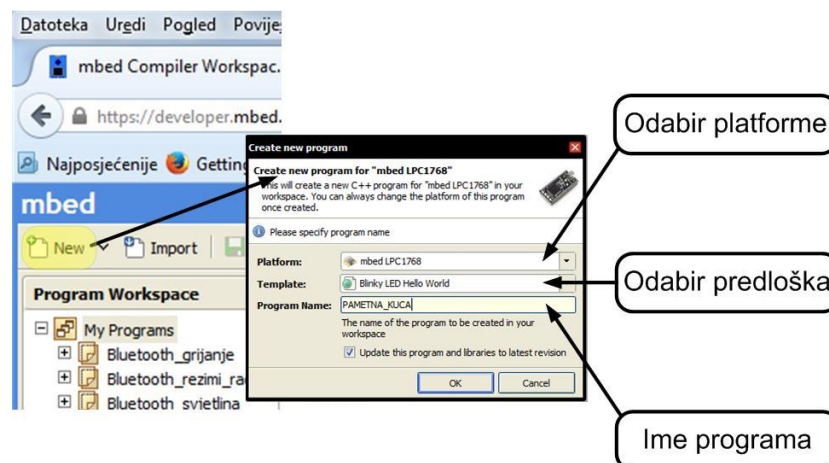
Slika 5.2. Početna stranica razvojnog okruženja *mbed* mikroračunala

Instalacija dodatnih programskih alata nije potrebna jer se programiranje i prevođenje programskog koda u strojni kod izvodi *online*. To znači da je postupak izrade programa brži, pošto je omogućena razmjena već postojećih programa i biblioteka između programera registriranih na *mbed* službenoj stranici. U gornjem desnom kutu početne *mbed* korisničke stranice nalazi se kartica *Compiler* kojom se pristupa prevodiocu. Prevodilac provjerava sintaksu programa i ne dopušta izvršavanje programa ako detektira grešku. Slika 5.3. prikazuje *mbed* prevodioca.



Slika 5.3. Mbed prevodioc

Programski kod pisan je u C++ programskom jeziku. Prije otvaranja novog projekta potrebno je odabrati platformu, odnosno vrstu *mbed* mikroračunala (u ovome slučaju LPC1768) koja će se koristiti pri programiranju. Odabir je bitan jer platforme nemaju jednak raspored pinova, stoga deklariranje ulaznih ili izlaznih pinova unutar koda može biti neispravno. Pritiskom na karticu u gornjem desnom kutu dodaje se *mbed* platforma/platforme koja/koje će se koristiti. Klikom na karticu *New* otvara se novi program (slika 5.4.).



Slika 5.4. Stvaranje novog programa u *mbed* razvojnom okruženju

Odabirom novog programa otvara se dodatan prozor u kojem se definira platforma koja će se trenutno koristiti, upisuje se ime programa i odabire između praznog i „*Blinky LED Hello World*“ predloška. Ispravnost programskog koda se provjerava pomoću kartice *Compile*. Ako prevodioc nije pronašao nikakvu grešku, programski kod se sprema u *.bin datoteku koja se pohranjuje na računalo. Datoteka s ekstenzijom *.bin sprema se na *mbed Flash* disk te se programiranje izvršava pritiskom tipke RESET.

5.1 Program upravljanja pametnom kućom

Sustav upravljanja pametnom kućom može se podijeliti na tri bitna podsustava:

1. upravljanje osvjetljenjem,
2. upravljanje i regulacija temperature i
3. upravljanje vratima (protuprovalna zaštita).

Navedena podjela je poslužila kao predložak pisanja programskog koda. Svaki podsustav čini tzv. prekidnu rutinu koja se pokreće svaki put kada se pomoću aplikacije na android mobilnom uređaju pošalje određena naredba putem *Bluetooth* komunikacije. Pozivanje prekidne rutine izvršava se unutar glavnog dijela programa gdje se uspoređuje dobiveni niz bajtova. Najprije se uspoređuje prvi bajt odnosno slovo koje okida određenu prekidnu rutinu. Sljedeća dva bajta određuju izvršavanje naredbe (npr. uključanje svijetla u boravku ili podizanje roleta) unutar određene prekidne rutine. U privitku moguće je naći kompletan programski kod *mbed* mikroračunala, potreban za upravljanje pametnom kućom [34], [35], [36], [37], [38], [39], [40], [41].

5.1.1 *Upravljanje osvjetljenjem*

Osim uključenja i promjene intenziteta pojedinog svjetla unutar kuće, rasvjeta je potrebna i izvana. Prekidna rutina *osvjetljenje()* izvršava naredbe uključenja svjetala ispred ulaznih i garažnih vrata. Da bi se mogao mijenjati intenzitet potrebno je definirati PWM izlaze unutar programskog koda. LPC1768 ima šest PWM mogućih izlaza smještenih na pinovima P21÷P26. Promjenom aktivnog ciklusa⁸ mijenja se intenzitet svjetla. Rolette su također dio podsustava osvjetljenja. Upravlja se preko koračnog motora, odnosno preko drivera za kojeg je bila potrebna biblioteka „*sMotor*“ dostupna na *mbed* službenoj stranici.

5.1.2 *Upravljanje i regulacija temperature*

Prekidna rutina *grijanje()* odnosi se na regulaciju temperature unutarnjeg prostora kuće. Preko aplikacije se šalje željena vrijednost temperature unutar kuće, koju se nastoji dobiti uz pomoć grijača, ventilatora i temperaturnog senzora kao povratne veze. Deklarirane su dvije varijable. Jedna za trenutnu vrijednost temperature, a druga se odnosi na traženu vrijednost koja je pristigla putem aplikacije. Unutar glavnog programa te dvije vrijednosti se uspoređuju i na temelju usporedbe uključuje se ili isključuje ventilator i grijač. Ako je izmjerena temperatura viša od tražene, uključit će se ventilator koji nastoji sniziti temperaturu. U slučaju da je izmjerena temperatura niža od tražene, grijač se mora uključiti kako bi došlo do porasta temperature unutar prostora.

5.1.3 *Upravljanje vratima*

Radi sigurnosti i protuprovale, vrata je potrebno zaključavati. Pomoću aplikacije moguće je upravljati elektromagnetskom bravom koja služi za zaključavanje i otključavanje ulaznih vrata. Bravom se upravlja preko releja. Osim ulaznih vrata model pametne kuće ima i garažna vrata upravljana preko koračnog motora.

5.1.4 *Režimi rada*

Ponekada je potrebno stvoriti određenu vrstu atmosfere i ugođaja unutar kuće. Istovremeno upravljanjem podsustava, na najbolji mogući način žele se ostvariti uvjeti prikladni za određenu vrstu događaja unutar kuće. Pametna kuća u ovom završnom radu ima pet mogućih režima rada detaljno opisanih u tablici 5.1.

⁸eng Duty cycle – 0.0 (0%) = najmanji intenzitet, svjetlo je ugašeno; 1.0 (100%) = najveći intenzitet.

Tablica 5.1. Režimi rada pametne kuće

Režim rada	Tražena temperatura	Osvjetljenje (intenzitet u %)			Ulazna vrata
		Boravak	Spavaća soba	Hodnik	
Isključeno/default	21.0°C	0%	0%	0%	Zaključana
Party	19.0°C	70%	80%	50%	Otključana
Romantika	25.0°C	30%	30%	30%	Zaključana
Nema me doma	Režim rada koji naizmjenično uključuje i isključuje svjetla kako bi se eventualno zavarao provalnik.				
Ured	22.0°C	100%	100%	100%	Zaključana

6. Izrada aplikacije za upravljanje pametnom kućom

Upravljanje pametnom kućom samome korisniku treba biti što jednostavnije. Stoga je potrebno osmisлити način kako uspostaviti komunikaciju između korisnika (osobe koja zadaje naredbe) i upravljačke ploče (sustava koji izvršava dobivene naredbe). Komunikacija je ostvarena pomoću *Android* mobilne aplikacije. Koristeći *Bluetooth*, korisnik preko aplikacije bežičnim putem upravlja modelom pametne kuće. U opisu *mbed* mikroračunala spominjao se pojam brze izrade prototipa (eng. *Rapid prototyping*) koji istovremeno traži razvijanje programskog koda i njegovo testiranje na mikroračunalu. Ovakav opis može se pripisati *App Inventor* programskome alatu za izradu *Android* aplikacije. To je *open source* web aplikacija koju razvija istraživačko sveučilište MIT za multinacionalnu korporaciju *Google*. Iako koristi JAVA-u kao programski jezik, *App Inventor* ne traži njegovo poznavanje. Programiranje je olakšano pomoću blokova koji predstavljaju određeni programski kod. Blokovi se dovode na radnu površinu programa i poput slagalice se spajaju tako da se dobije željena funkcija u aplikaciji. Pristup alatu se obavlja prijavom ili registriranjem vlastitog *Google* korisničkog računa na stranici <http://appinventor.mit.edu/explore/>. Razvojno okruženje *App Inventor*a se može podijeliti na dva dijela:

1. dizajner komponenata (*Component designer*) – zadužen za vizualni dio aplikacije i
2. uređivač blokova (*Block editor*) – zadužen za funkcionalni dio aplikacije.

6.1 Dizajner komponenata

Postupak izrade aplikacije započinje definiranjem njezinog izgleda. Raspored vidljivih i nevidljivih komponenata te njihova pozicija na ekranu *Android* uređaja se uređuje unutar dizajnera komponenata. Slika 6.1. prikazuje izgled dizajnera komponenata koji se može podijeliti na pet segmenata. Sve moguće komponente koje se mogu iskoristiti za izradu aplikacije smještene su na paleti (eng. *Palette*). Paleta je podijeljena na nekoliko sekcija. Na slici 6.1. prikazuje najčešće korištenu *User Interface* sekciju koja sadrži: klizače⁹, gumbove¹⁰, tekstualne oznake¹¹, potvrдне¹², tekstualne

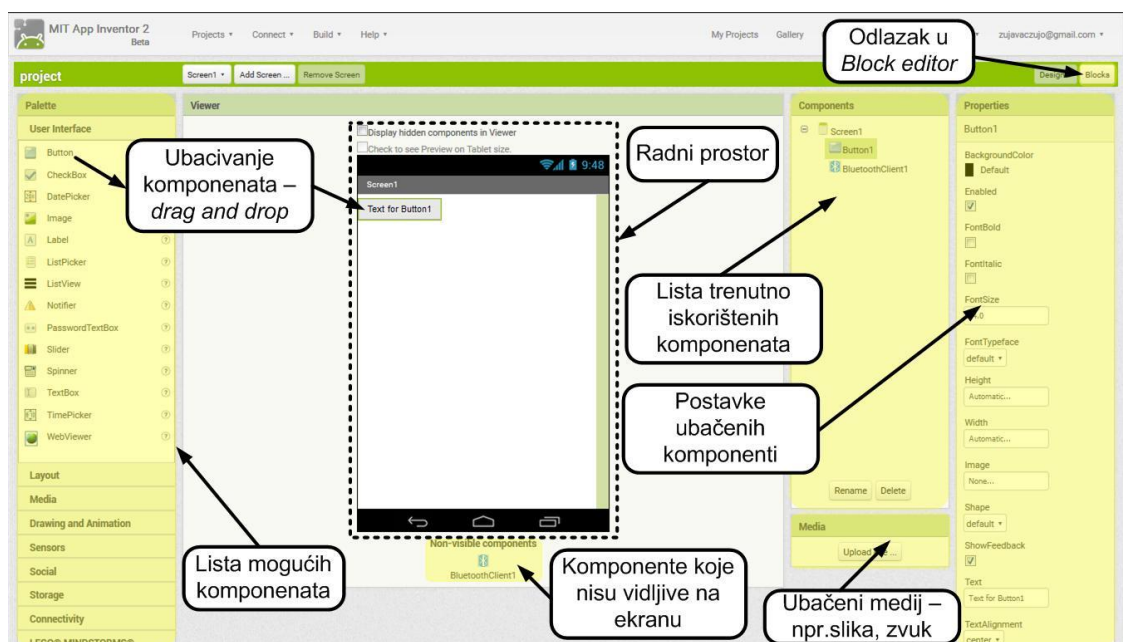
⁹ eng. *slider*.

¹⁰ eng. *button*.

¹¹ eng. *label*.

¹² eng. *check box*.

okvire¹³ itd. Osim navedenih komponenti za izradu aplikacije korištene su komponente iz skupina *Layout*¹⁴ i *Connectivity*¹⁵.



Slika 6.1. Dizajner komponentata

6.2 Uređivač blokova

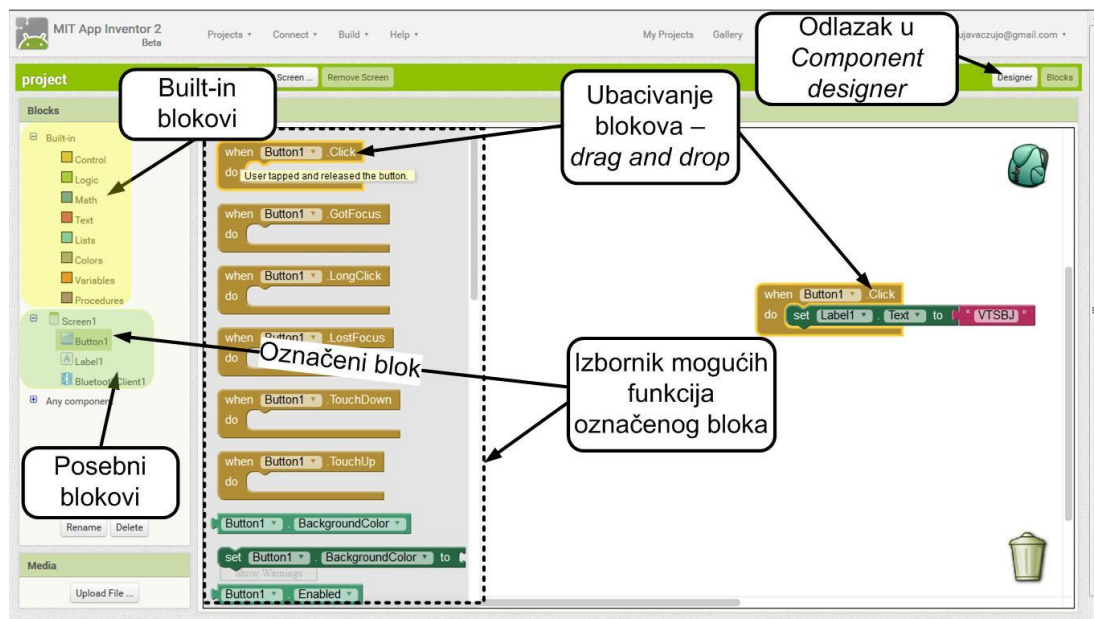
U desnome kutu razvojnog okruženja, nalaze se dvije kartice: *Designer* i *Block*. Klikom na *Block* karticu radnja se prebacuje u uređivač blokova (eng. *Block editor*). Zadaća uređivača blokova je dodjeljivanje naredbi komponentama koje su ubačene u dizajneru komponentata. Kombinacijom blokova kreiraju se funkcije potrebne za izvršavanje zadataka aplikacije. Najbitnije je da se pomoću blokova početnicima olakša stvaranje aplikacije iako ne poznaju programski jezik čije se znanje traži u ostalim programskim alatima za izradu aplikacije. Slika 6.1. prikazuje uređivač blokova. Unutar uređivača definira se funkcionalnost komponente koja je ubačena u dizajneru komponentata. Postoje dvije kategorije blokova:

- 1) *Built-in* blokovi – uvijek su dostupni i
- 2) posebni blokovi – dostupni tek kada se ubaci pojedina komponenta unutar dizajnera.

¹³ eng. *text box*.

¹⁴ *Layout* – pomaže oko rasporeda i preglednosti komponentata na ekranu.

¹⁵ *Connectivity* – sadrži komponente koje se tiču povezivanja android uređaja na *Internet* ili pomoću *Bluetootha*

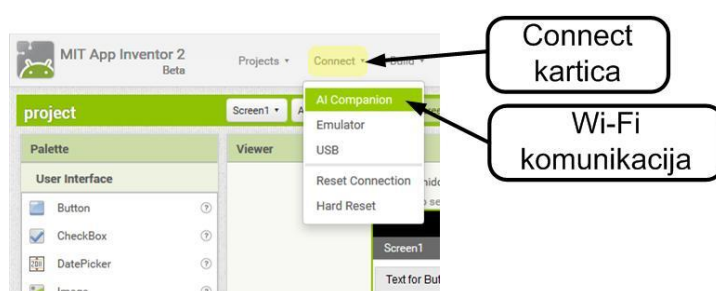


Slika 6.1. Block editor

6.3 Testiranje aplikacije

Prilikom izrade, aplikaciju je moguće istovremeno testirati iako nije do kraja izvedena. Dodatna pogodnost je emulator, koristan za programere koji nemaju vlastiti *Android* uređaj. Odabirom kartice *Connect* (slika 6.2.) smještene na alatnoj traci nude se tri načina testiranja:

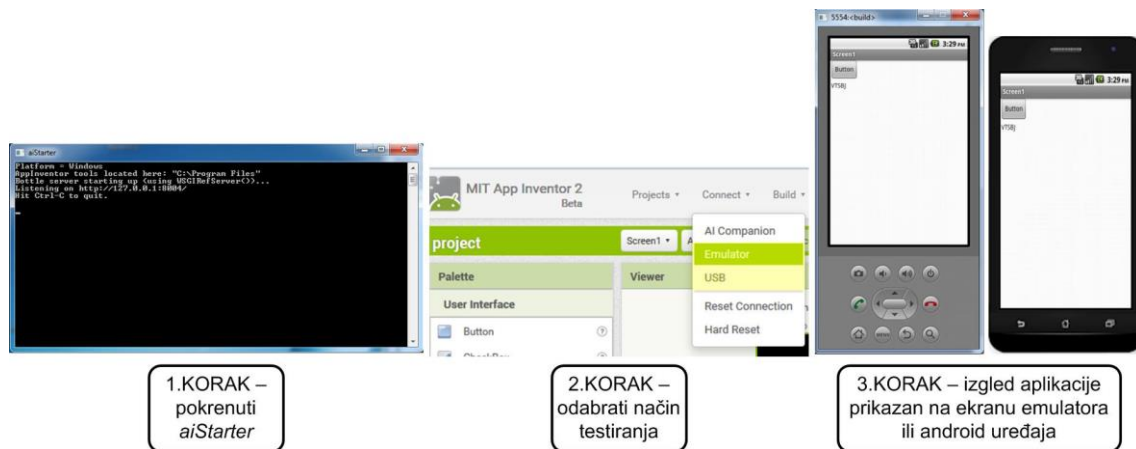
1. putem emulatora,
2. preko USB veze ili
3. preko *Wi-Fi* mreže.



Slika 6.2. Vrste testiranja aplikacije

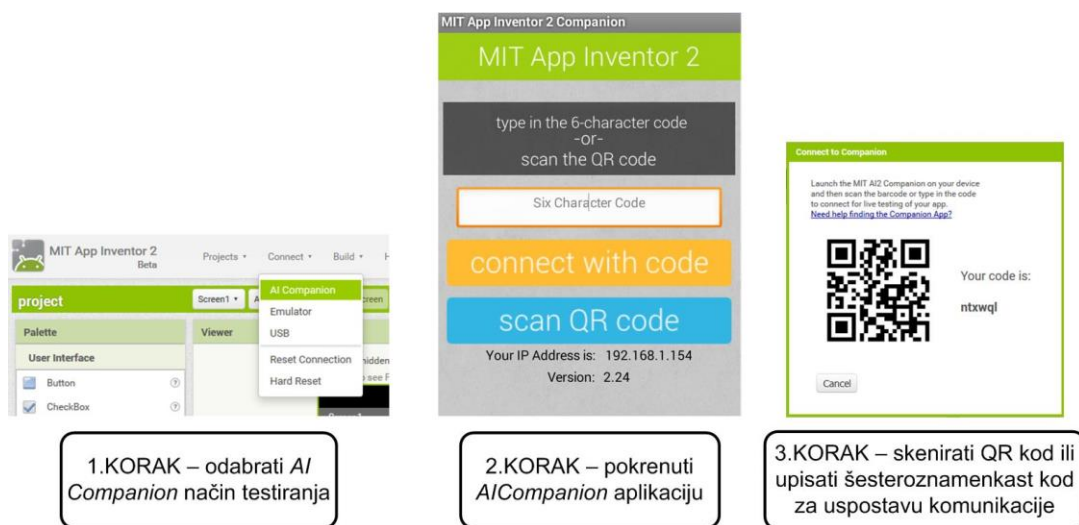
Emulator predstavlja virtualni *android* mobilni uređaj na ekranu računala. Da bi internetski preglednik pravilno komunicirao s emulatorom, potrebno je instalirati *aiStarter* program, besplatan i dostupan na službenoj stranici *App Inventor*-a. Program

je potreban i kod USB testiranja. Slika 6.3 prikazuje postupak testiranja pomoću emulatora ili preko USB veze.



Slika 6.3. Testiranje aplikacije – emulator i USB veza

Korisnici koji imaju vlastiti android uređaj testiranje mogu obaviti koristeći *AICompanion* aplikaciju. Ako je dostupan pristup *Wi-Fi* bežičnoj mreži odabirom *AI Companion* unutar kartice *Connect* korisnik dobiva *QR* kod¹⁶ kojeg je potrebno skenirati. Skeniranjem koda omogućuje se pristup aplikaciji koja se trenutno programira (slika 6.4.). *AICompanion* aplikacija potrebna je i koda testiranja putem USB kabela. Korištenjem *WiFi* mreže potrošnja baterije uređaja može biti veća nego uobičajena. Testiranje USB vezom uklanja navedeni problem jer se istovremeno komunikacija i punjenje uređaja obavlja preko kabela [42], [43].



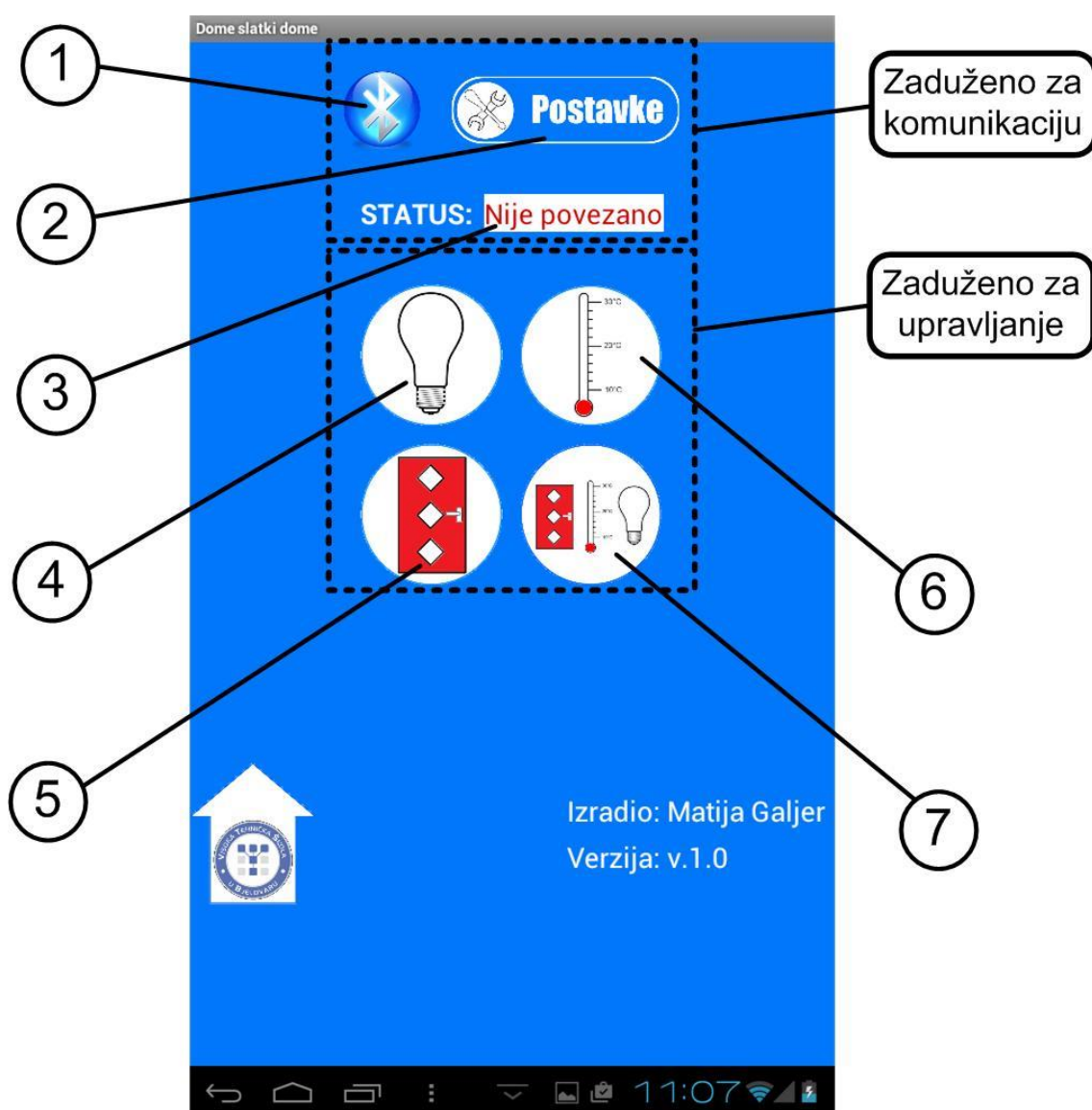
Slika 6.4. Testiranja aplikacije – *Wi-Fi*

¹⁶ eng. *Quick Response* kod

6.4 Aplikacija za upravljanje pametnom kućom

Korisnik upravlja pametnom kućom pomoću aplikacije *PametnaKuca* čiji se programski kod nalazi u privitku. Aplikacija omogućuje da se putem *Bluetooth* komunikacije šalju naredbe *mbed* mikroračunalu koji ih obrađuje i izvršava. Izgled ekrana aplikacije nakon njezina pokretanja prikazan je na slici 6.5. Opis komponenti prikazan je na tablici 6.1. Ekran je podijeljen na dva dijela:

1. dio zadužen za *Bluetooth* komunikaciju i
2. upravljački dio.



Slika 6.5. Izgled početnog ekrana aplikacije

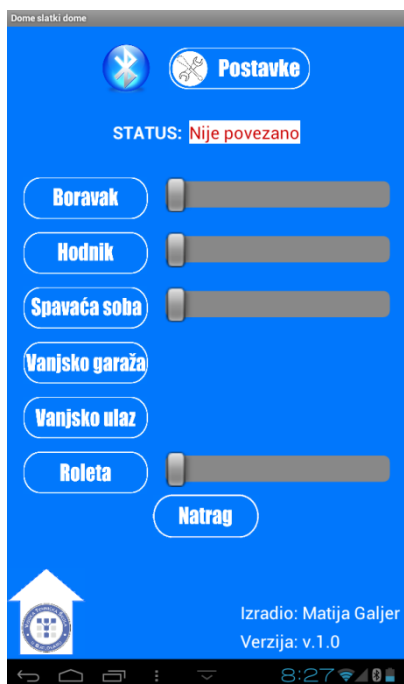
Tablica 6.1. Opis komponenata aplikacije

Broj na slici 6.5	Naziv komponente	Vrsta komponente	Opis/zadaca komponente
1.	„Bluetooth“	<i>List picker</i> ¹⁷	Prikazuje listu mogućih <i>Bluetooth</i> uređaja za povezivanje
2.	„Postavke“	<i>Button</i>	Prikazuje postavke <i>android</i> uređaja
3.	„Status“	<i>Label</i>	Prikazuje dali je <i>android</i> uređaj spojen/odspojen od bluetooth modula
4.	„Osvjetljenje“	<i>Button</i>	Upravljanje osvjetljenjem unutar kuće
5.	„Vrata“	<i>Button</i>	Otključavanje/zaključavanje ulaznih i garažnih vrata
6.	„Temperatura“	<i>Button</i>	Upravljanje temperaturom unutar kuće
7.	„Režimi rada“	<i>List picker</i>	Prikazuje listu mogućih režima rada. Elementi liste: „Isključeno/default“, „Party“, „Romantika“, „Nisam doma“, „Romantika“.

Pokretanjem, aplikacija traži dopuštenje za aktivacijom *Bluetooth* komunikacije ako nije bila uključena u postavkama. Pritiskom na *Bluetooth list picker* otvorit će se popis adresa svih uređaja koji sadrže *Bluetooth*. Broj komponenti upravljačkog dijela aplikacije jednak je broju prekidnih rutina programskog koda *mbed* mikroračunala.

¹⁷ *List picker* – vrsta *buttona* koji korisniku nudi listu tekstualnih elemenata na odabir. Elementi se definiraju unutar postavki komponente.

Odabirom pojedine komponente otvara se dodatan niz komponenti zaduženih za upravljanje pojedinog dijela kuće (slika 6.6.).



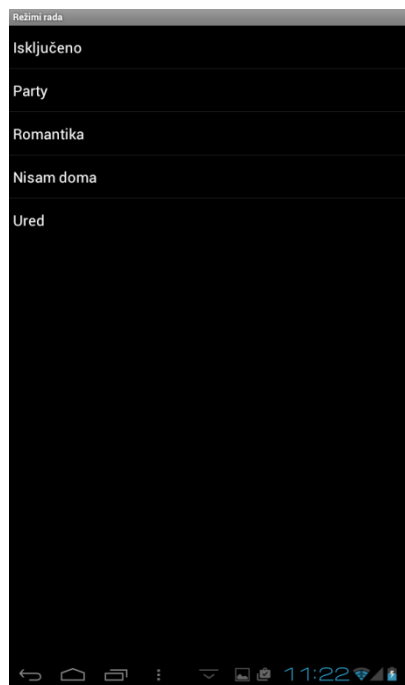
a) „Osvjetljenje“



b) „Temperatura“



c) „Vrata“



d) „Režimi rada“

Slika 6.6. Upravljački dio aplikacije za pojedinu prekidnu rutinu

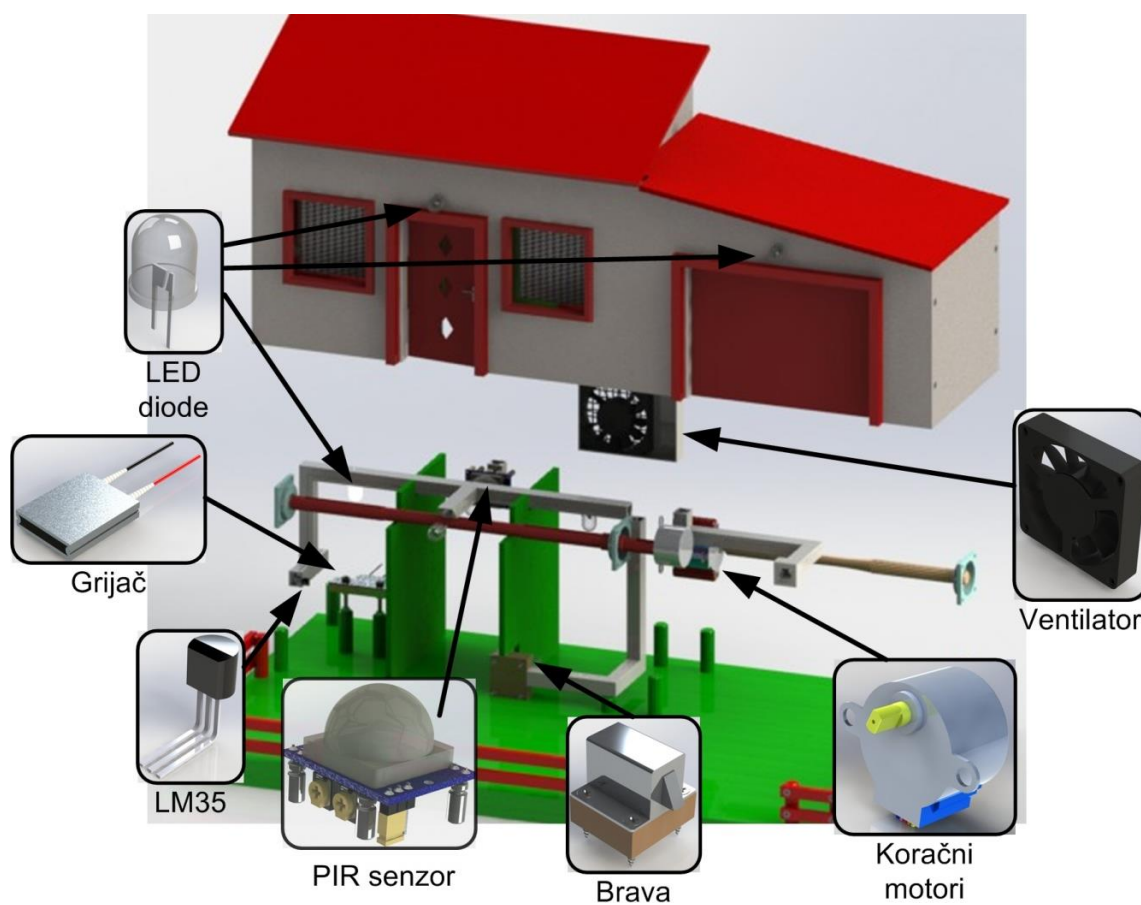
Pritiskom na „Osvjetljenje“ otvoriti će se novi ekran zadužen za upravljanje roletama i LED diodama unutar pojedine prostorije. Tipka „Grijanje“ nudi mogućnost upisa željene temperature unutar kuće. Tipka „Vrata“ zadužena je za otključavanje/zaključavanje ulaznih vrata i za upravljanje garažnih vrata. „Režimi rada“ nudi parametriranje elemenata unutar kuće kao što je prikazano u prethodnome poglavlju unutar tablice 5.1.

7. Prikaz rada pametne kuće

Osnovni aktuatori pametne prikazani su u tablici 7.1. Raspored aktuatora detaljno je prikazan na slici 7.1.

Tablica 7.1. Aktuatori pametne kuće i njihova osnovna namjena

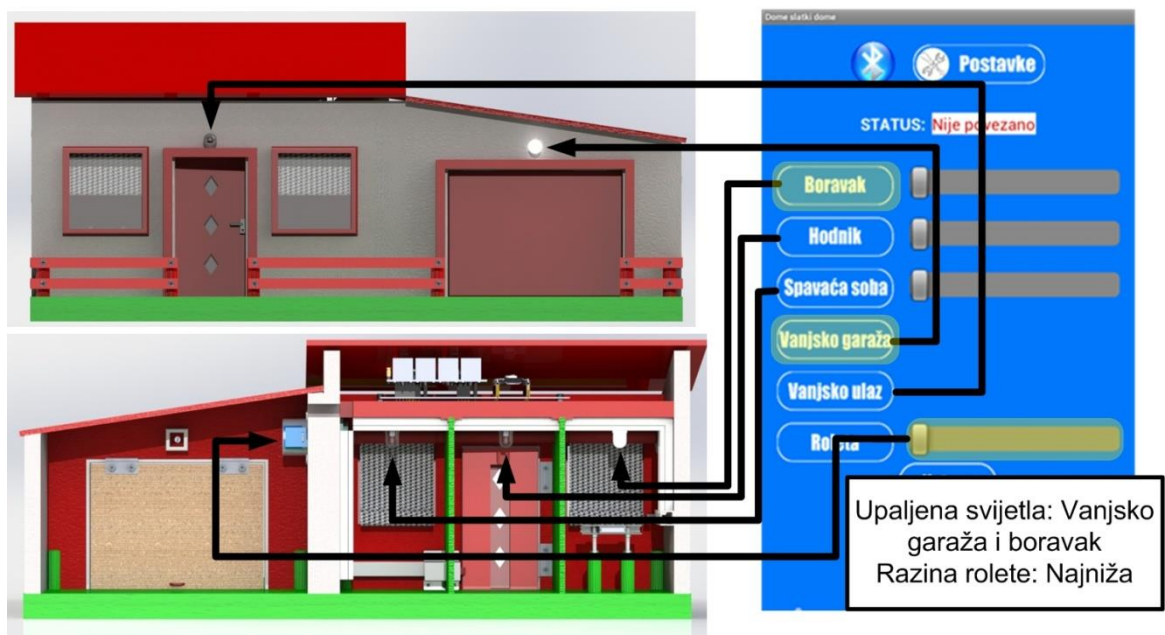
Aktuatori	Namjena
1. LED diode	Osvjetljenje
2. Koračni motori	
3. Temperaturni senzor (LM35)	Regulacija temperature
4. Grijač	
5. Ventilator	
6. PIR senzor	Protuprovalna zaštita
7. Brava	



Slika 7.1. Raspored aktuatora na modelu pametne kuće [40]

7.1 Upravljanje osvjetljenjem modela pametne kuće

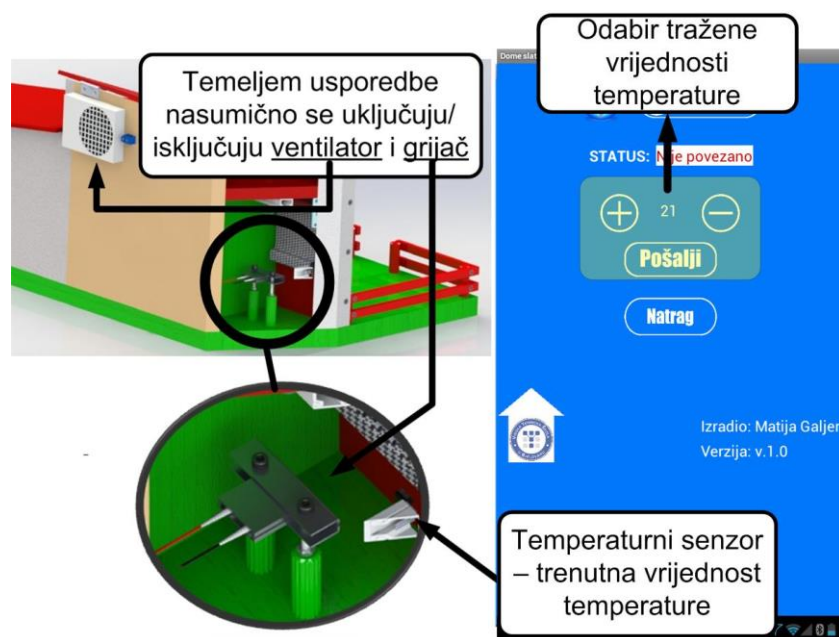
Osvjetljenje se upravlja promjenom inteziteta pojedine LED diode te razinom podizanja/spuštanja roleta. Aplikacijom je moguće mijenjati intezitet nad tri LED diode (u boravku, hodniku i spavaćoj sobi). Ostale dvije LED diode (vanjska garažna i kućna LED dioda) zadužene su za osvjetljavanje vanjskog prostora. Razina roleta mijenja se upravljanjem koračnog motora. Slika 7.2. prikazuje poprečni presjek modela pametne kuće gdje se mogu vidjeti navedene komponente zadužene za upravljanje osvjetljenjem.



Slika 7.2. Upravljanje osvjetljavanjem

7.2 Prikaz regulacije temperature unutar modela pametne kuće

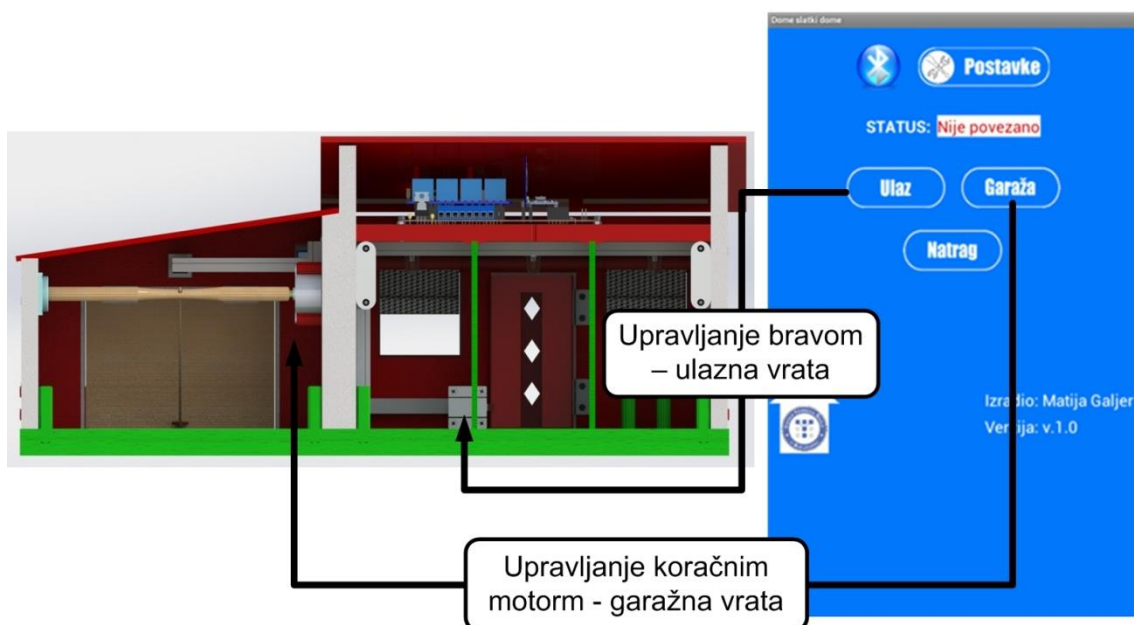
Regulacija temperature izvršava se na temelju usporedbe željene vrijednosti temperature poslane preko *Android* aplikacije s trenutnom vrijednošću temperature dobivene preko temperaturnog senzore LM35. Slika 7.3. prikazuje osnovne komponente i princip rada regulacije temperature.



Slika 7.3. Princip rada regulacije temperature

7.3 Prikaz rada protuprovalne zaštite modela pametne kuće i upravljanje vratima

Upravljanje vratima ujedno služi kao protuprovalna zaštita pametne kuće. Aplikacijom upravlja se sa bravom čija je funkcija zaključavanje ulaznih vrata kuće. Također aplikacijom se upravlja sa garažnim vratima, odnosno koračnim motorom pomoću kojeg se vrata otvaraju. Slika 7.4. prikazuje model pametne kuće s osnovnim komponentama zaduženim za protuprovalnu zaštitu.



Slika 7.4. Princip rada protuprovalne zaštite

8. Zaključak

Danas je nemoguće zamisliti život bez moderne tehnologije. Njenim naglim razvojem postala je sve zastupljenija i jeftinija. Upravo je cijena tehnologije od presudne važnosti i ključ tehnološke globalizacije. Tehnologija se više ne mora povezivati s luksuzom. Pametna kuća danas više ne mora biti luksuz, već se pomoću *Android* uređaja i mikroračunala relativno jeftino može modernizirati vlastiti dom. U ovom radu pametna kuća upravlja se pomoću *mbed* mikroračunala koji dobiva naredbe od korisnika preko *Bluetooth* aplikacije dizajnirane u *App Inventor*u. Mikroračunala sadržavaju sva potrebna sučelja za upravljanje i komunikaciju te istovremeno imaju brzu reakciju na tražene zahtjeve. Navedene prednosti pogodne su za brzu izradu prototipa. Brzo eksperimentiranje i testiranje nudi *App Inventor*. Prilagođen je korisniku koji nema dovoljno znanja oko programiranja u JAVA programskome jeziku kojeg zamjenjuju tzv. blokovi naredbi. Komunikacija između mikroračunala i aplikacije se ostvaruje pomoću *Bluetooth* komunikacije čiji je domet nekoliko desetaka metara što je dovoljno za manje stambene objekte. Sustav pametne kuće podijeljen je na tri podsustava: sustav upravljanja osvjetljenjem (upravljanje roletama i rasvjetom), sustav za upravljanje i regulaciju temperature i sustav protuprovale (upravljanje vratima).

9. Literatura

- [1] https://en.wikipedia.org/wiki/X10_%28industry_standard%29 (dostupno: 20.10.2015.)
- [2] http://www.alibaba.com/product-detail/x10-smart-dimmer-lamp-module-home_716883956.html (dostupno 20.10.2015.)
- [3] <https://en.wikipedia.org/wiki/CEBus> (dostupno 20.10.2015.)
- [4] <http://www.hometoys.com/content.php?url=/htinews/aug97/articles/kwacks/kwacks.htm> (dostupno 20.10.2015.)
- [5] https://en.wikipedia.org/wiki/KNX_%28standard%29 (dostupno 20.10.2015.)
- [6] <http://www.intrahouse.eu/en/products/smart-home-solutions/intrahouse-knx/> (dostupno 20.10.2015.)
- [7] <https://en.wikipedia.org/wiki/ZigBee> (dostupno 20.10.2015.)
- [8] http://www.aliexpress.com/store/product/CC2530-Zigbee-Module/604119_579637698.html (dostupno 20.10.2015.)
- [9] http://www.digiwireless-solutions.com/Bizit/store/product.php?id_product=180 (dostupno 20.10.2015.)
- [10] <https://grabcad.com/library/mbed-nxp-lpc1768-1> (dostupno 28.3.2016.)
- [11] <https://grabcad.com/library/passive-thd-1> (dostupno 28.3.2016.)
- [12] <https://grabcad.com/library/78xx-1> (dostupno 28.3.2016.)
- [13] Ljubivoj Cvitaš, Brzi razvoj prototipova na bazi mikroupravljača, Tehničko veleučilište u Zagrebu, 2014. (dostupno: 25.8.2015.)
- [14] <https://developer.mbed.org/platforms/mbed-LPC1768/> (dostupno 25.8.2015.)
- [15] <https://developer.mbed.org/cookbook/4-Phase-Step-Motor> (dostupno 7.9.2015.)
- [16] <http://www.hztk.hr/media/NMT2015./PRIPREMA9-2015.pdf> (dostupno 7.9.2015.)
- [17] <http://www.instructables.com/id/BYJ48-Stepper-Motor/> (dostupno 20.10.2015.)
- [18] <https://hr.wikipedia.org/wiki/Otpornik> (dostupno 20.10.2015.)
- [19] <http://www.ebay.com/itm/Aluminum-PTC-Heater-Thermostat-Heating-Plate-28-x-20-x-5mm-220V-160C/231660821828?hash=item35f00f9d44:g:aO8AAOSw3ydV1pNG> (dostupno 20.10.2015.)
- [20] <http://www.ebay.co.uk/itm/40mm-12V-2-Pin-Computer-Cooler-Cooling-Fan-PC-Black-C-/130576851023> (dostupno 20.10.2015.)

- [21] <http://www.ebay.com/itm/DC12V-Cabinet-Door-Electric-Lock-Assembly-Solenoid-Drawer-Sauna-Luggage-Locker-/361328755928?hash=item5420df44d8:g:dpcAAOSw~gRVi7Dk> (dostupno 24.11.2015.)
- [22] http://4tronix.co.uk/store/index.php?rt=product/product&product_id=153 (dostupno 24.11.2015.)
- [23] <http://products.li2.in/45-pir-sensor.html> (dostupno 20.10.2015.)
- [24] <http://en.nysenba.com/index.php/list-5-1.html> (dostupno 20.10.2015.)
- [25] http://www.electfreaks.com/wiki/index.php?title=PIR_Motion_Sensor_Module:DYP-ME003 (dostupno 20.10.2015.)
- [26] <http://www.datasheetspdf.com/PDF/HC-SR501/775435/2> (dostupno: 15.9.2015.)
- [27] <https://en.wikipedia.org/wiki/Bluetooth> (dostupno: 23.9.2015.)
- [28] <http://www.ebay.com/itm/HC-05-Wireless-Bluetooth-RF-Transceiver-Module-serial-RS232-TTL-for-arduino-/281667025902?hash=item4194a9bbec:g:eVkAAOSwcwhVM84X> (dostupno: 24.11.2015.)
- [29] <http://www.ebay.com/itm/Brand-New-USB-2-0-to-TTL-UART-5PIN-Module-Serial-Converter-CP2102-STC-PRGMR-/400735354008?hash=item5d4db03c98:g:hMkAAOSwDk5T4eSL> (dostupno: 24.11.2015.)
- [30] <http://www.instructables.com/id/Arduino-AND-Bluetooth-HC-05-Connecting-easily/> (dostupno: 7.9.2015.)
- [31] <http://www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/?ALLSTEPS> (dostupno: 7.9.2015.)
- [32] http://www.linotux.ch/arduino/HC-0305_serial_module_AT_comamnd_set_201104_revised.pdf (dostupno: 7.9.2015.)
- [33] <http://forum.pcekspt.com/showthread.php?t=14622> (dostupno 20.10.2015.)
- [34] <https://developer.mbed.org/components/PIR-Motion-Sensor/> (dostupno: 15.9.2015.)
- [35] <https://developer.mbed.org/cookbook/Bluetooth-Android-Controlled-MBED> (dostupno: 23.9.2015.)

- [36] <https://developer.mbed.org/users/yoonghm/notebook/digital-output/> (dostupno: 23.9.2015.)
- [37] <http://mitappsinventor.blogspot.in/2015/05/control-multiple-servo-motors-using.html> (dostupno: 23.9.2015.)
- [38] <https://developer.mbed.org/users/MadVoltage/notebook/mbed-lpc1768-eagle-library/> (dostupno: 23.9.2015.)
- [39] https://developer.mbed.org/users/GRG/code/LM35/docs/aed747a24dcd/main_8c_pp_source.html (dostupno: 23.9.2015.)
- [40] <https://developer.mbed.org/cookbook/4-Phase-Step-Motor> (dostupno: 28.9.2015.)
- [41] https://developer.mbed.org/users/lasmahei/code/StepperMotor_28BYJ-48/
- [42] David Wolber, Hal Abelson, Ellen Spertus, Liz Looney, App Inventor2, Create Your Own Android Apps, O'Reilly Media Inc., Sebastopol Kalifornija, 2015. (dostupno: 29.8.2015.)
- [43] Paula Beer, Carl Simmons, Hello App Inventor! Android programming for kids and the rest of us, Manning Publications Co., New York, 2015. (dostupno: 29.8.2015.)

Sažetak

Naslov: Upravljanje pametnom kućom pomoću *mbed* mikroračunala

U ovom završnom radu opisan je koncept pametne kuće. Model pametne kuće sastoji se od kuće kao stambenog dijela i garaže. Opisan je postupak izrade tiskane pločice od samog projektiranja u programskome alatu *Eagle* do kemijske i mehaničke obrade. Detaljno su opisane korištene upravljačke komponente te njihove funkcije. Objašnjen je princip rada programa *mbed* mikroračunala. Također je opisan programski alat *APP Inventor* za izradu *Android* aplikacije te njezin opis i *Bluetooth* veza pomoću koje komunicira s *mbed* mikroračunalom.

Ključne riječi: pametna kuća, *Bluetooth*, *Eagle*, *mbed*, *App Inventor*.

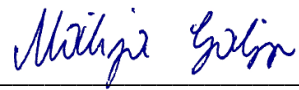
Abstract

Title: Control of smart home using mbed microcontrollers

In this final thesis the concept of smart house is described. Smart home model includes a residential section and garage. The procedure of making printed circuit board is described, from designing in programming tool Eagle to chemical and mechanical processing. Used control components and drivers are described in detail and also their main functions. Final thesis explains the work principle of mbed microcontroller program code. For creating the android applications programming tool App Inventor is described. App Inventor helped to develop the control application which we are using for controlling the mbed microcontroller via Bluetooth communication.

Keywords: smart home, Bluetooth, Eagle, mbed, App Inventor.

Završni rad izrađen je u Bjelovaru, 8.4.2016.

A handwritten signature in blue ink, reading "Matija Galin". The signature is written in a cursive style with a large initial 'M'.

(Potpis studenta)

Prema Odluci Visoke tehničke škole u Bjelovaru, a u skladu sa Zakonom o znanstvenoj djelatnosti i visokom obrazovanju, elektroničke inačice završnih radova studenata Visoke tehničke škole u Bjelovaru bit će pohranjene i javno dostupne u internetskoj bazi Nacionalne i sveučilišne knjižnice u Zagrebu. Ukoliko ste suglasni da tekst Vašeg završnog rada u cijelosti bude javno objavljen, molimo Vas da to potvrdite potpisom.

Suglasnost za objavljivanje elektroničke inačice završnog rada u javno dostupnom nacionalnom repozitoriju

Matija Galjer

(Ime i prezime)

Dajem suglasnost da se radi promicanja otvorenog i slobodnog pristupa znanju i informacijama cjeloviti tekst mojeg završnog rada pohrani u repozitorij Nacionalne i sveučilišne knjižnice u Zagrebu i time učini javno dostupnim.

Svojim potpisom potvrđujem istovjetnost tiskane i elektroničke inačice završnog rada.

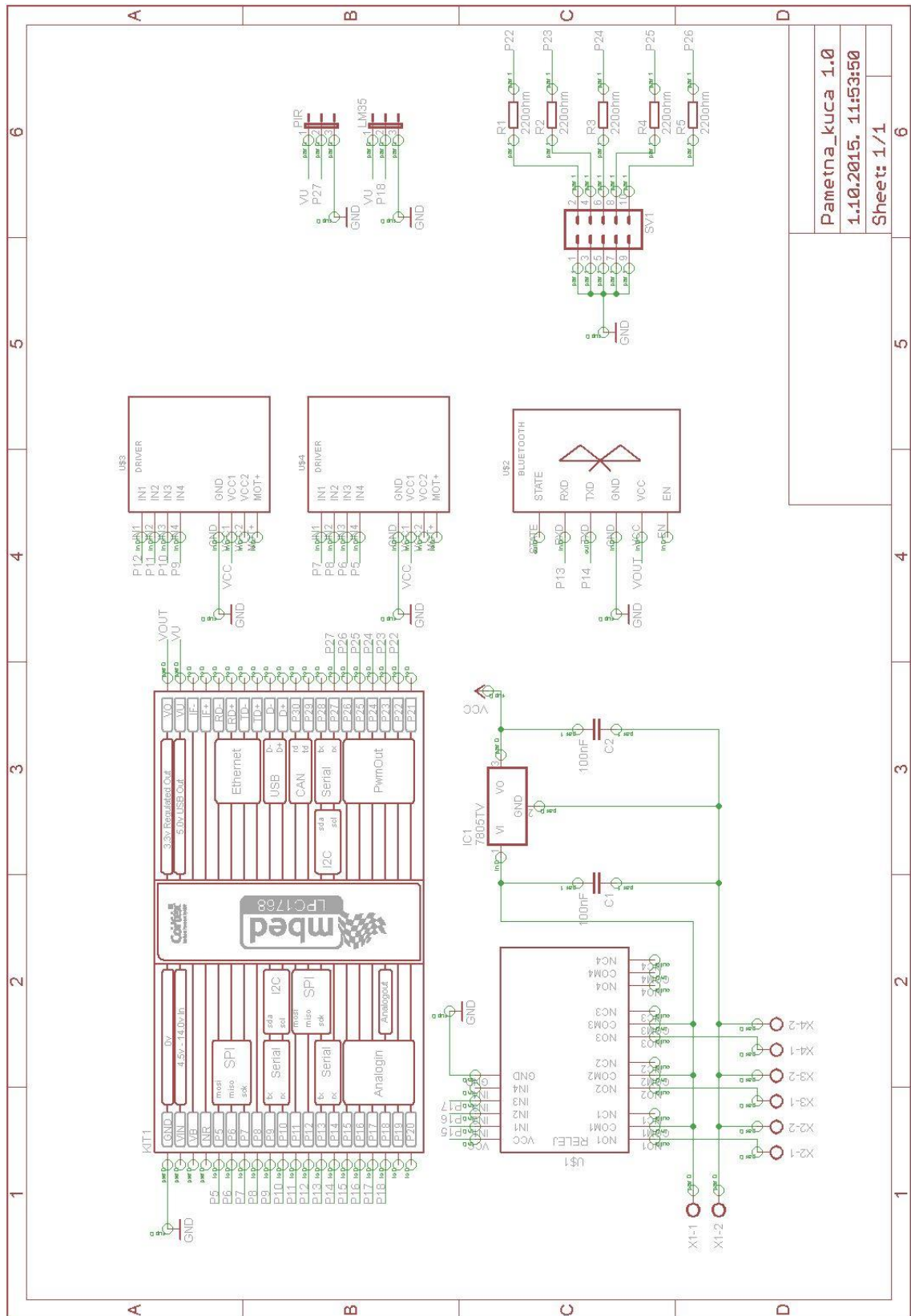
U Bjelovaru, 27.4.2016.



(potpis studenta/ice)

Privitak

A) Shematski prikaz tiskane pločice



B) Programski kod – *mbed*

```
#include "mbed.h"
#include "sMotor.h"

Serial pc(USBTX,USBRX); //definiranje
serijske komunikacije izmedju racunala i mbed-a
Serial HC05(p13,p14); //definiranje
serijske (Bluetooth) komunikacije izmedju mbed-a i HC05 bluetooth modula

sMotor roleta(p7,p8,p6,p5); //definiranje
upravljackih pinova (IN1,IN2,IN3,IN4) za driver ULN2003 za upravljanje
roletom
sMotor garaza(p12,p11,p10,p9); //definiranje
upravljackih pinova (IN1,IN2,IN3,IN4) za driver ULN2003 za upravljanje
garaznim vratima
DigitalOut brava(p15,1); //0= ZATVORENO,
1= OTVORENO, spojen na IN1 upravljacki pin releja
DigitalOut grijac(p16); //spojen na IN2
upravljacki pin releja
DigitalOut ventilator(p17); //spojen na IN3
upravljacki pin releja
PwmOut boravak(p22);
PwmOut hodnik(p23);
PwmOut spavaca(p24); //spavaća soba
PwmOut vanjsko(p25); //vanjsko
ulazno svijetlo
PwmOut vanjsko2(p26); //vanjsko
svijetlo garaza
InterruptIn PIR(p27); //definiranje
ulaznog pina za senzor pokreta
AnalogIn LM35(p18); //definiranje
analognog ulaznog pina za temperaturni senzor LM35

unsigned int tx = 48; //varijabla
koja se šalje Android uređaju kada PIR senzor detektira pokret
unsigned char rx; //varijabla
koja pohranjuje znak dobiven preko bluetootha
unsigned int rx1; //varijabla
koja pohranjuje prvi bajt dobiven preko bluetootha
unsigned int rx2; //varijabla
koja pohranjuje drugi bajt dobiven preko bluetootha
int vrijeme; //varijabla u
koju se pohranjuje vrijednost timera
int prazna_kuca = 0; //flag koji
pokreće režim rada "Nema me doma"
int ukupno; //varijabla
koja pohranjuje zbroj 2 bajta
int brzina_koraka = 1800; //brzina (u us)
promjene koraka
int broj_koraka;
int broj_koraka1 = 0;
int broj_koraka2;
int ukupno1 = 0;
int smjer_roleta; //0= u smjeru
kazaljke na satu; 1= u smjeru obrnutom od kazaljke na satu
int gar_vrata = 1; //0= kada su
otvorena; 1= kada su zatvorena
int brojac = 0;
int brojac2 = 0;
float svjetlina = 0.0;
float svj_postotak; //jacina
svijetlosti u postotcima
float ref_temp;
//referentna/trazena temperatura
float izmj_temp; //varijabla
koja pohranjuje izmjerenu temperaturu preko LM35 senzora
```

```

void osvjetljenje(){
    pc.printf("'Osvjetljenje interrupt'\n\r");
    pc.printf("Ukupno: %d\n\r", ukupno);
    switch(ukupno){
        case 1:
            //Boravak_button u aplikaciji
            boravak =! boravak;
            break;
        case 2:
            //Hodnik_button u aplikaciji
            hodnik =! hodnik;
            break;
        case 3:
            //Spavaca_button u aplikaciji
            spavaca =! spavaca;
            break;
        case 4:
            //Vanjsko_ulaz_button u aplikaciji
            vanjsko =! vanjsko;
            break;
        case 5:
            //Vanjsko_garaza_button u aplikaciji
            vanjsko2 =! vanjsko2;
            break;
    }
    if(ukupno>=1000 && ukupno<=1100){
        //Boravak_slider u aplikaciji
        svj_postotak = (ukupno%1000);
        svjetlina = svj_postotak/100;
        boravak = svjetlina;
        pc.printf("Svjetlina boravak: %1.2f %\n\r", svj_postotak);
    }
    else if(ukupno>=2000 && ukupno<=2100){
        //Hodnik_slider u aplikaciji
        svj_postotak = (ukupno%2000);
        svjetlina = svj_postotak/100;
        hodnik = svjetlina;
        pc.printf("Svjetlina hodnik: %1.2f %\n\r", svj_postotak);
    }
    else if(ukupno>=3000 && ukupno<=3100){
        //Spavaca_slider u aplikaciji
        svj_postotak = (ukupno%3000);
        svjetlina = svj_postotak/100;
        spavaca = svjetlina;
        pc.printf("Svjetlina spavaca soba: %1.2f\n\r", svj_postotak);
    }
    else if(ukupno>=4000 && ukupno<=4512){
        //Rolete_slider u aplikaciji
        if(ukupno > ukupno1){
            smjer_roleta = 1;
        }
        else {
            smjer_roleta = 0;
        }
        broj_koraka2 =(ukupno%4000);
        broj_koraka = abs(broj_koraka2 - broj_koraka1);
        pc.printf("Broj koraka: %d\n\r Broj koraka1: %d\n\r", broj_koraka,
        broj_koraka1);
        roleta.step(broj_koraka,smjer_roleta,brzina_koraka);
        ukupno1 = ukupno;
        broj_koraka1 = broj_koraka2;
    }
}

void grijanje(){
    pc.printf("'Grijanje interrupt'\n\r");
    ref_temp = ukupno;
}

```

```

        pc.printf("Zeljena temperatura: %d stupnjeva C\n\r", ukupno);
        pc.printf("Izmjerena temperatura: %1.1f stupnjeva C \n\r", izmj_temp);
    }

    void rezimi_rada(){
        pc.printf("'Rezimi rada interrupt'\n\r");
        pc.printf("Rezim rada: ");
        switch(ukupno){
            case 1: //rezim_rada =
ISKLJUCENO/DEFAULT
                ref_temp = 21.0;
                boravak = 0;
                hodnik = 0;
                spavaca = 0;
                brava = 1;
                pc.printf("ISKLJUCENO/DEFAULT\n\r");
                prazna_kuca = 0;
                break;
            case 2: //rezim_rada =
PARTY
                ref_temp = 19.0;
                boravak = 0.7;
                hodnik = 0.5;
                spavaca = 0.8;
                brava = 1.0;
                pc.printf("PARTY\n\r");
                prazna_kuca = 0;
                break;
            case 3: //rezim_rada =
ROMANTIKA
                ref_temp = 25.0;
                boravak = 0.3;
                hodnik = 0.3;
                spavaca = 0.3;
                brava = 0.0;
                pc.printf("ROMANTIKA\n\r");
                prazna_kuca = 0;
                break;
            case 4: //rezim_rada =
NISAM DOMA
                prazna_kuca = 1;
                pc.printf("NEMA ME DOMA\n\r");
                break;
            case 5: //rezim_rada =
URED
                ref_temp = 22.0;
                boravak = 1.0;
                hodnik = 1.0;
                spavaca = 1.0;
                brava = 0;
                pc.printf("URED\n\r");
                prazna_kuca = 0;
                break;
        }
    }

    void vrata(){
        if(ukupno == 1){
            brava = ! brava;
        }
        else if(ukupno == 2){
            garaza.step(1800,gar_vrata,brzina_koraka); // broj koraka
= 128 (cetvrtina kruga)
        }
        gar_vrata = ! gar_vrata;
    }
    void uljez(){

```

```

        if(HC05.writeable()){
            HC05.putc(tx);
            pc.printf("Saljem uljez na andorid");
        }
    }

int main() {
    HC05.baud (38400); //baud rate
    Bluetooth komunikacije
    set_time(1256729737);
    while(1) {
        svjetlina = 0.0;
        izmj_temp = LM35.read(); //ocitava
        analognu vrijednost napona temperaturnog senzora
        izmj_temp = izmj_temp*2.9*100; //formula za
        pretvaranje analogne vrijednosti u stupnjeve celzijuse
        if(HC05.readable()){
            rx = HC05.getc();
            rx1 = HC05.getc();
            rx2 = HC05.getc();

            ukupno = ((rx2*256)+rx1);
            pc.printf("Primljeno %c %d %d\n\r", rx, rx1, rx2);

            switch(rx){
                case 103: //ASCII g = 103
                    pc.printf("GRIJANJE: \n\r");
                    grijanje();
                    break;
                case 114: //ASCII r = 114
                    pc.printf("REZIMI RADA: \n\r");
                    rezimi_rada();
                    break;
                case 115: //ASCII s = 115
                    pc.printf("SVJETLINA: \n\r");
                    osvjetljenje();
                    break;
                case 118: //ASCII v = 118
                    pc.printf("VRATA: \n\r");
                    vrata();
                    break;
            }
        }
        if(ref_temp < izmj_temp){
            grijac = 1;
            ventilator = 0;
        }
        else if(ref_temp > izmj_temp){
            grijac = 0;
            ventilator = 1;
        }
        else {
            grijac = 0;
            ventilator = 0;
        }
        if (prazna_kuca == 1){
            do {
                time_t seconds = time(NULL);
                if(HC05.readable()){
                    rx = HC05.getc();
                }
                PIR.rise(&uljez);
                if(seconds % 5 == 0){
                    brojac++;
                }
                else if (seconds % 3 == 0){
                    brojac2++;
                }
                else if (brojac == 2){

```

```

        hodnik != hodnik;
        brojac = 0;
    }
    else if (brojac2 == 5){
        spavaca != spavaca;
        brojac2 = 0;
    }
}
while (rx != 1);
ukupno = rx;
prazna_kuca = 0;
rezimi_rada();
}
}

```


C) Programski kod – App Inventor

